IEEE Big Data 2022
This is a self-archived pre-print version of this article.
The final publication is available at IEEE via
http://dx.doi.org/10.1109/BigData55660.2022.10020266.

Adaptive Attacks and Targeted Fingerprinting of Relational Data

Tanja Šarčević
SBA Research, Vienna, Austria
tsarcevic@sba-research.org

Rudolf Mayer
SBA Research, Vienna, Austria
rmayer@sba-research.org

Andreas Rauber
Vienna University of Technology, Austria
rauber@ifs.tuwien.ac.at

***Abstract*—Fingerprinting is a method of embedding a traceable mark into digital data to (i) verify the owner and (ii) identify the recipient of a released copy of a data set. This is crucial when releasing data to third parties, especially if it involves a fee, or if the data is of sensitive nature and further sharing and leaks should be discouraged and deterred from. A fingerprint is required to (i) be robust against modifications to the data to achieve successful ownership protection, while (ii) affecting the quality and utility of the data as little as possible.**

So far, literature mostly assumes *attackers with rather limited capabilities* who perform random modification to the dataset. With a certain task in mind to perform on the data, the attacker can however perform an adaptive and targeted attack that maximises its chances of removing or invalidating the fingerprint, while reducing the data utility the least. In the same line, the data owner can *optimise the robustness* of the scheme by anticipating a specific focus of the attacker and focusing the fingerprint embedding on the most valuable parts of the data. In this paper, we, therefore, provide an in-depth discussion on threat models, targeted attacks and adaptive defences. We further demonstrate the impact of targeted attacks on classical and, in comparison, adaptive fingerprinting in an empirical manner.

***Index Terms*—Fingerprinting, Intellectual Property Protection, Adaptive Attacker, Machine Learning, Data utility**

I. Introduction

Digital fingerprinting is an information-hiding method that helps to protect intellectual property for various types of data. By combining and embedding a secret, owner- and recipient-specific mark into the data, fingerprinting allows to identify (i) the source of digital objects and (ii) the source of an unauthorised data leakage. It thus removes barriers and facilitates sharing data with third parties, where different recipients of the data obtain differently marked content. Since fingerprinting does not control access to the data, it is considered a *passive* (reactive) protection tool.

A fingerprint in the domain of relational (tabular) data is often realised by a pseudo-random pattern of modifications within the values of the dataset that can be embedded and extracted using only the (owner's) secret key (Figure 1). The quality of such fingerprinting methods generally comprises of two complementary aspects: (i) the (remaining) **utility** of the fingerprinted data and (ii) the **robustness** of the fingerprinting scheme. Data utility is inevitably decreased by fingerprinting, as it introduces modifications into the data. The utility is in most cases measured by observing changes in statistical moments of the data, e.g. mean or variance (*data-oriented*), but can also be done in a *use-oriented* fashion, when the data purpose is known or can be assumed, e.g. the data set is the training set for a predictive Machine Learning (ML) task. The robustness of a fingerprinting scheme is defined as its resilience against modifications to the fingerprinted data. Modifications manifest either as a consequence of benign updates on the dataset or as malicious attacks. An attack is a collective notion of different types of attempts to prevent the correct detection of a fingerprint. To this end, an attacker might modify, delete or add values to the fingerprinted data to modify or erase the fingerprint. These attacks are often considered to be random and *uninformed*, and thus generally result in an additional decrease in data utility — therefore a fingerprint is considered robust if it cannot be removed without significantly reducing data utility, which would render the attacked data of little value to the attacker.

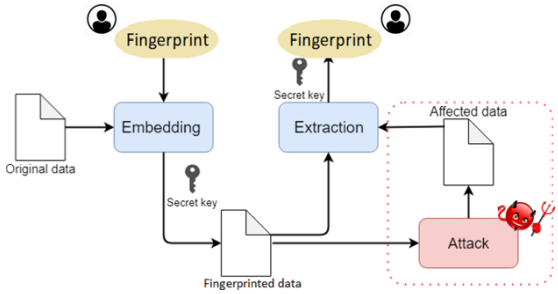


Fig. 1. Fingerprinting process

Several different fingerprinting schemes have been proposed in literature [1], [2]. While they are generally evaluated in terms of the robustness and utility, we identify **several shortcomings in the overall evaluation process**.

Firstly, the robustness analysis in related works lacks a discussion on data utility decrease due to the attacks. Stronger attacks are usually achieved by applying more modifications to the original fingerprinted data, hence the attempts to remove the fingerprint come at a cost, which we call the *attacker's*

This work was partially funded by the "Industrienahe Dissertationen" program (No 878786) of the Austrian Research Promotion Agency (FFG) under the project "IPP4ML" and European Union's Horizon2020 research and innovation programme under grant agreement No 826078 (project FeatureCloud). This publication reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.
978-1-6654-8045-1/22/$31.00 ©2022 IEEE

cost. To quantify this cost, a good robustness evaluation should contain an evaluation of the amount of utility the attacker loses due to their attack, which is our first research question:

RQ1: How to measure the cost of attacks, i.e. the utility loss which comes as a consequence of malicious attacks on fingerprinted data?

Secondly, most related works consider a form of an attacker that is assumed to have rather **limited capabilities, and mostly performs random modifications to the dataset**, e.g. random bit flips or random deletion of rows. However, we do argue that there is a lack of comprehensive threat analysis and model. Namely, with a certain task in mind to perform on the data, for example, learning a predictive ML model, the attacker can perform a *targeted* attack that reduces data utility the least w.r.t. the task (i.e. with the lowest attacker's cost). For example, the attacker can try to remove the fingerprint by removing those attributes from the dataset that contribute least to the task (i.e. perform some form of feature selection). Hence, we focus on the research questions:

RQ2: What are targeted, utility-aware attacks that the threat model can be extended with?

RQ3: How successful are the targeted attacks in removing the fingerprint while acceptably preserving data utility?

In the same line as an attacker can optimise the attack, the data owner can optimise the robustness by focusing on the most valuable portion of the data. Along the example above, the data owner might embed the fingerprint into the data not in a completely random way, but focus on the values that are of most interest to preserve. Regarding the defence, we focus on the following research questions:

RQ4: How can we defend against targeted attacks?

In our work, we answer *RQ1* by addressing the **attacker's utility loss** as a measured performance loss for an assumed predictive ML task on the data, and we incorporate it in the robustness analyses throughout this paper. Observing only the change in statistical moments may not cover complex cases when evaluating the attacker's utility loss; when, for example, an attacker deletes a large number of random rows intending to remove the fingerprint, these statistical moments might change only marginally (in line with the law of large numbers), while the utility of learning a predictive model might plummet. To answer *RQ2* and *RQ3*, we then propose a **targeted attack model**, where the attacker, with a specific task in mind, optimises their malicious modification such that the utility is preserved. For that, we adapt well-known attacks and obtain a (i) *targeted horizontal subset attack* by heuristic-based undersampling method and (ii) *targeted vertical subset attack* by feature selection method. We evaluate the attacker's utility loss when using the targeted attacks and show that this approach can indeed be more successful, especially for targeted vertical subset attacks.

To defend against potential targeted vertical attacks, we propose an **adaptive fingerprinting** process that utilises a feature selection method for the choice of attributes that should contain most of the fingerprint marks and thus address *RQ4*. We also discuss how choosing the right parameters for fingerprinting intrinsically leads to better robustness, which hence should be the first step in preventive defence.

Our main contributions are, therefore:

- A notion of attacker's utility loss as an integral part of the robustness analysis
- An adaptive, targeted attacker model, aiming to attack the scheme in the most cost-efficient way
- An adaptive fingerprinting as a defence focusing on the most valuable assets and mitigating targeted attacks
- An empirical evaluation of the targeted attacks and adaptive fingerprint

The remainder of this paper is organised as follows: Section II provides the background on fingerprinting relational data and discusses related work. In Section III, we propose our targeted attacker model to extend the currently existing threat model against fingerprinting, and evaluate it in Section IV. In answer to the targeted attacks, we propose adaptive fingerprinting as a defence in Section V. Finally, Section VI provides conclusions and an outlook on future work.

II. Background and related work

Watermarking and fingerprinting techniques were first developed for the multimedia domain [3]. The generally large amount of data required to represent this content (e.g. images or video) offers sufficient space to embed the marks without significantly affecting the actual content. The application of fingerprinting and watermarking was later extended to other types of digital data, where the effects caused by marking are of a bigger concern. These types of content include e.g. text [4], software [5], graphs [6], sequential data [7], and relational databases [8].

A. Fingerprinting Schemes for Relational Data

A fingerprinting scheme in principle encompasses two main processes – embedding and detection. These are shown in Figure 1. As proposed by Li et al. in a pioneering scheme for fingerprinting relational data [8], within the embedding process, the fingerprint, a bit string that is unique for each recipient, is first created as a function of the owner's secret key and the recipient's unique identifier, usually using a hash function. Secondly, the fingerprint is embedded into the data as a pattern of modifications made to data values. These steps are the same for schemes proposed later, e.g. [9]–[12] (for numerical data) and [13]–[15] (categorical), while the pattern itself depends on the scheme. For example, in [11], the data is first divided into several blocks, each of which is associated with one fingerprint bit. The authors of [12] propose a pattern where the fingerprint bits are embedded in two separate phases. In all cases, the patterns depend on pseudo-random number generation, seeded by the owner's secret key. This ensures that only with access to the secret key, the pattern can be recreated, but not otherwise, even if the algorithm steps of the embedding scheme are known.

Fingerprint detection is the process complementary to embedding. Using the secret key and the fingerprinted dataset, the marks are decoded into a fingerprint uniquely attributable

to a data recipient. The fingerprinted dataset may be subject to certain benign modifications or malicious attacks, which may affect the detection algorithm, therefore one needs to ensure the scheme is robust by design. Further, the fingerprinted data set is shared with recipients that want to e.g. use it for data analysis, and therefore its utility needs to be preserved, despite the modifications.

Table I presents several *parameters* that describe properties of and control the behaviour of a fingerprinting scheme, i.e. the way the fingerprint is created and embedded into the dataset.

TABLE I
FINGERPRINTING PARAMETERS

Parameter	Description
n_marks	abs. or rel. number of marks in the dataset
n_attr	number of attributes chosen for marking
$magnitude$	eg. for numerical values the number of least significant bits available for marking
L	length of a fingerprint
N	number of recipients
τ	the "certainty threshold" that is used to decide whether a (partially) extracted fingerprint bit is accepted
ω	"redundancy factor" that estimates how many times each of the L fingerprint bits is embedded into the data

B. Data utility

Modifications introduced by data fingerprinting unavoidably change the data, and thus in most cases reduce the utility of the data; therefore, modifications should be minimised in the marking process. Too large modifications can further affect the perceptibility of the fingerprint mark and facilitate its removal.

In literature, most of the early works claim preservation of data utility simply as a consequence of the design of the fingerprinting scheme, which ensures that the modifications are made only on the least significant bits of the candidate values distributed sparsely in the data set [11], [12]. Later works then consider two main groups of utility measures, as described in the following: (i) *data-oriented* and (ii) *use-oriented* metrics.

a) Data-oriented utility: (or mark perceptibility) is described as a measure of change in mean and variance of numerical attributes of a fingerprinted data set [8], [16]. The statistical changes in the datasets due to fingerprinting can be generalised to a class of utility measures capturing the general similarity of fingerprinted data to their original, referred to as *data-oriented utility metrics*.

b) Use-oriented utility: (or task-oriented utility) is measured on a specific scenario of usage of the data. For example, the utility can be measured as the effect of the modifications that were introduced by the fingerprint on a data analysis conducted on fingerprinted data. e.g. database querying or a predictive modelling task.

Some schemes address the notion of utility in a known data purpose with the design of their scheme [9], [13]. For example, the data holder may define a set of utility constraints for their dataset, such as preserving the quality of certain queries that are especially sensitive and require bounding the amount of error. Satisfying the usability constraints on the scheme design level is relatively cumbersome, as the recipient needs to manually define the constraints depending on their intended data use. Furthermore, there must be a mutual consensus between the data holder and the recipient to avoid an overly restrictive set of constraints that do not allow applying a robust fingerprint.

For the purpose of using fingerprinted data as a training set for a ML model, the informative use-oriented utility metrics are those capturing the impact that the fingerprint has on the effectiveness of performing the task, i.e. comparing common metrics such as accuracy, F1 score, mean squared error, etc. [17]. Unlike satisfying query constraints during the fingerprint embedding process, the loss in ML performance cannot be easily bounded at the embedding phase, due to the stochastic nature of many modelling techniques.

C. Robustness

The robustness of a fingerprinting scheme is measured as the resilience of the scheme against modifications of the data set; modifications can happen as a result of benign updates (e.g. data scaling), or malicious attacks. The resilience manifests as the success of the detection scheme to extract the fingerprint from a data copy and associate it with its correct recipient. In literature, frequently mentioned attacks against fingerprinting schemes for relational datasets include [2]:

- *subset attack*: the attacker releases only a portion of the fingerprinted data: either a subset of tuples (records, rows) in a *horizontal* subset attack, or a subset of attributes (features, columns, ...) in a *vertical* subset attack,
- *flipping attack (alteration attack)* or modifying the least-significant bits of data values,
- *attribute-level alteration attack* [1],
- *superset attack*, i.e. a record-insertion attack,
- *additive attack* where a fake fingerprint is embedded "on top" of the existing and
- *correlation attack* [18] where an adversary uses prior knowledge of the database to modify suspicious values due to high correlations between certain columns or rows.

Robustness is usually expressed via *false miss* (fm), i.e. the inability to detect the correct fingerprint, either due to a wrongly detected candidate, or no candidate at all. The embedding randomness of the fingerprinting schemes proposed for relational data allows a good statistical estimation of the above measures for the *naïve* attacks (c.f. Section III-A) [8], [11] which closely matches the results from an empirical robustness analysis [17].

In literature, an attack is generally considered successful when it causes a high fm rate while not rendering the data useless for the intended purpose of the attacker, measured by not distorting the data too much with the attack. There is, however, a limited discussion about the attacker's utility loss. The attack strength in robustness evaluations (i.e. number of modifications) is bounded by a value determined by a rule of thumb. For example, the authors in [8] consider modifying up to 50% of values in a flipping attack and in [9] argue that removing more than 80% of rows by a subset attack is

an undesirably large portion of data. However, it is hard to generalise what is the amount of data that the attacker would consider worth publishing, since, for example, 20% of very large data sets still results in many published records, and might almost perfectly preserve data-oriented utility metrics such as mean values, or be sufficient for downstream use-oriented metrics, e.g. obtaining a model with very similar accuracy. Nevertheless, beyond the number of modifications, no analysis addresses the cost of attack in terms of utility.

III. Threat Model and Robustness of Fingerprinting

A fingerprinting scheme instance is considered robust if the attacker cannot remove the fingerprint from the data set without substantially reducing the data's utility. To estimate the remaining usefulness of the data for the attacker, we compute the utility loss via the use-oriented metrics described in Section II-B. In this section, we first classify attacker models and describe the approach for robustness estimation and attacker's utility loss evaluation. Secondly, we propose novel, targeted attacks, which we expect to be a bigger threat compared to the well-known attacks from literature described in Section II-C.

A. Attacker models

Attacking a fingerprinting process may be done via multiple channels, for example modifying the fingerprinted data to confuse the fingerprint detection process, attacking the execution of the embedding or detection process, accessing the data holder's secret key, etc. Additionally, within these channels, there is an overwhelming amount of possible actions for an attacker. In this work, we restrict the definition of an attacker on two fronts: (i) in regards to scheme access, we consider a *white-box access* and *grey-box access* attacker and (ii) in regards to data background knowledge we differentiate between a *naïve attacker* and a *targeting attacker*.

a) White-box access: The attacker is assumed to know the algorithmic steps of the embedding and detection processes, and all fingerprinting parameters, such as length of a fingerprint, strength and magnitude. Only the owner's secret key remains unknown to the attacker. The attacker does not have access to the execution of either the embedding or detection process. The white-box attacker can thus be considered well informed about the fingerprinting scheme. Note that white-box access does not help the attacker in determining the mark locations, since they are done randomly using the owner's secret key.

b) Grey-box access: The grey-box access applies in the adapted fingerprinting setting introduced later in Section V. This type of access restricts the knowledge of the additional methods that are used in the fingerprinting process and their parameters, e.g. the method to select data columns to be marked, and the number of marked columns. The grey-box access otherwise resembles the white-box access.

c) Naïve attacker: This attacker does not use any background knowledge about the data set nor its intended use, and all the modifications (flipping, deletion, etc.) are applied randomly to the data values, i.e. each value has an equal probability to be attacked. This is the type of attacker that related work until now considers.

d) Targeting attacker: The attacker uses background knowledge about the data or the predefined data purpose to reduce the utility loss caused by the attack. The attacker aims to perform stronger attacks (i.e. more modifications that distort the fingerprint) in a way that reduces utility less compared to a fully random set of modifications, e.g. by targeting specific attributes or rows that contribute less to the utility.

By combining these notions, we obtain three attacker models that we use and compare throughout the paper: a **naïve white-box attacker** (or just naïve attacker for brevity), **targeting white-box attacker** and **targeting grey-box attacker**.

B. Robustness definition and estimation

The robustness estimation in our process is focused on the *false miss* metric ($fm \in [0, 1]$). Robustness is empirically evaluated by recording the detection rate of the scheme instances under the attacks. A higher fm indicates stronger attacks.

To define the *robustness* of a scheme instance, a general measure that describes the resilience of the scheme instance against a specific attack, we introduce the following notation and definitions:

- $attack\_strength \in [0, 1]$ quantifies the amount of modification introduced by the attack with respect to the data size (e.g. percentage of rows deleted in horizontal subset attack, percentage of values flipped in the flipping attack, etc.)
- $tolerance \in [0, 100]\%$ denotes the "amount of mistakes" that the fingerprint detection process is allowed in the empirical analysis to be still considered successful. We use $tolerance$ to relax the notion of $robustness$ because even very high detection rates below 100% indicate a robust fingerprinting scheme instance. $tolerance$ is in our experiments set to 5%.

The $robustness \in [0, 1]$ of a scheme instance against an attack type is then the maximum $attack\_strength$ for which $fm < tolerance$, i.e. the most that the attacker can modify while the scheme instance keeps the detection rate failure up to the tolerance level. E.g. a $robustness = 0.80$ for a given scheme instance and dataset against a horizontal subset attack where $tolerance = 5\%$ means that the scheme successfully detects the fingerprint with at least 95% probability (i.e. fails up to 5% of times), even if the attacker deletes up to 80% of the rows of the dataset. Deleting more than 80% of data rows would in this case result in $\geq 5\%$ chance for detection failure.

C. Utility loss due to the attack

Attacking is no free lunch for an attacker either – any modification, i.e. the attack, on a released dataset is likely going to reduce the utility of the dataset. The expected reduction

in utility serves as a lower boundary on the robustness of the scheme instance to a data holder. For example, deleting 99% of the data rows will most likely remove the fingerprint, however, the data will likely be rendered useless – therefore to preserve the utility of the data, the attacker is limited in regards to the number of modifications.

To show the effects of the attacks on data utility, the utility loss needs to be observed on attacks that successfully remove the fingerprint – this represents the minimum utility loss that the attacker will obtain. The use-oriented metrics measuring ML effectiveness, described in Section II-B, are a good fit for estimating the attacker's utility loss because they highlight the differences between a naïve and targeting attacker, as the latter relies on having a defined data purpose (e.g. a predictive task). Furthermore, there is a lack of data-oriented utility metrics that can be used for the vertical subset attack – since data-oriented metrics are generally computed per attribute, for the vertical subset attack, which removes whole attributes, there are no fitting metrics that can capture data utility loss.

D. Targeted attacks

The attacks performed by the white-box targeting attacker are possible when the data has an assumed purpose, such as being used as a training set for a predictive ML model. In this scenario, the attacker can focus on creating a successful attack that at the same time preserves more utility than a random attack, for the given task. In particular, the attacker may modify some well-known attack types and utilise the use-oriented metrics to their advantage. We propose and evaluate two different types of targeted attacks: feature selection as a targeted vertical subset attack, and heuristic-based under-sampling as a targeted horizontal attack.

1) Targeted vertical subset attack (feature selection): Feature selection [19] is a well-known method used in ML processes, mostly for more efficiently (i.e. with less computational overhead) creating a predictive model by filtering the most relevant features for the predictive task[1]. Effectively, feature selection is a vertical subset attack, because entire columns are removed from the (fingerprinted) data set. An attacker, however, has in this case a much better chance of preserving use-oriented data utility compared to a naïve attacker, as the least important attributes are removed first. The success of such an attack depends primarily on the data itself, i.e. to which extent feature selection is possible without compromising too much of the model performance.

2) Targeted horizontal subset attack (under-sampling): The adapted version of the horizontal subset attack may be achieved by under-sampling the data set in a more informed way than random under-sampling, e.g. by trying to preserve the original distribution as much as possible. In ML processes, under-sampling is frequently performed on a training data set that is imbalanced with respect to the target attribute [20]. In a classification setting, this means that the size of the majority class(-es) is reduced, while samples from minority classes are kept in the data set. The goal of under-sampling is to improve the accuracy of the predictive model on the minority classes. While this can also be the goal of the attacker in scenarios with imbalanced data, the general scenario that we are interested in is that the attacker wants to publish a subset of data that preserves the original distribution as much as possible. For this, various under-sampling methods were proposed, usually based on a data sample distances, optimised by a heuristic method [21].

[1]Feature selection may also lead to more effective models, i.e. with higher accuracy or on a similar metric, if the removed features are noisy or redundant, and have no positive impact on the predictive power of the model.

IV. Evaluation

In this section, we evaluate the proposed targeting attacks from Section III-D against Li's fingerprinting technique [8][2]. We test the hypothesis of whether the targeting attacks result in decreased attacker's loss compared to the naïve attacks. For that, we fist obtain the baseline robustness and attacker's loss by applying naïve attacks to the fingerprinting scheme, followed by evaluating the attacker's loss after the corresponding targeting attacks. The results are obtained for vertical and horizontal subset attacks. The targeted horizontal attack is achieved by the *near-miss under-sampling* method that uses heuristic rules to select samples [21]. The targeted vertical attack relies on *impurity (gini index)*, i.e. the importance of a feature is represented by the mean decrease in the impurity of that feature[3].

The overview and the sequence of the methodology steps of evaluation are shown in Figure 2. In step 1), the data is fingerprinted using a combination of the presented values for percentage of marks and attributes, resulting in $20 \times 5 = 100$ different parameter settings, from which each is used 100 times for fingerprinting with a different random seed (influencing the fingerprint position) to achieve a robust estimate of fingerprint detection rate and the effects on utility.

In the second step of Figure 2, utility loss introduced by fingerprinting data is evaluated as a loss in the accuracy of ML models trained on the two datasets. Our results are obtained by training a set of different types of models and then especially focused on the best-performing ones given that the data purpose is well defined. This yields the most relevant and realistic insights into the attacker's utility loss. The ML performance evaluation employs 5-fold cross-validation, where the train set of each fold is the fingerprinted data rows and the test set are the original data rows. To evaluate the robustness in step 3, we attack each fingerprinted data such that we start with the weakest attack (5% for horizontal subset and flipping, and one column for vertical subset attack) and increase the attack strength until the detection algorithm fails to attribute the fingerprint to the correct recipient. This is the detection success boundary for the scheme with certain parameter setting, i.e.

[2]We adapt the technique originally proposed only for numerical data, so that, if a categorical value is chosen for marking, it is flipped to a random value from the attribute domain.

[3]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn.ensemble.GradientBoostingClassifier.feature_importances_

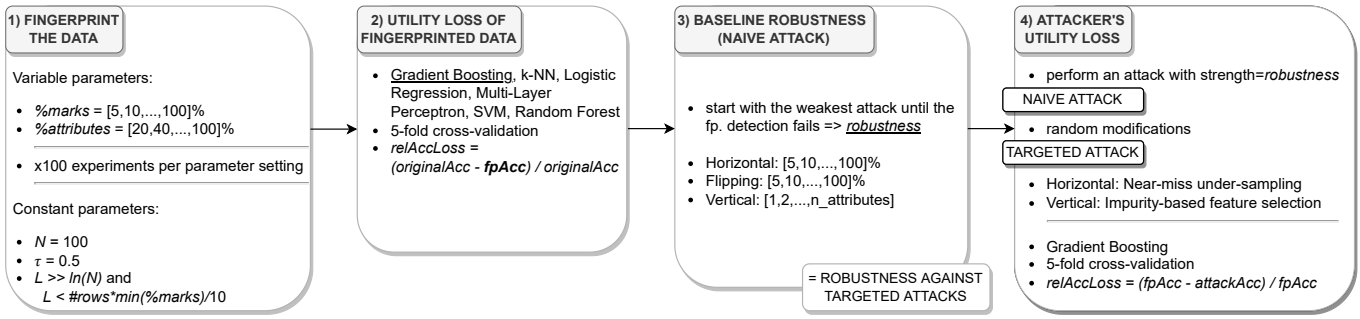


Fig. 2. Evaluation methodology workflow (for notation, cf. Table I)

the $robustness$ value (re. Section III-B). Robustness against the targeting attack has the same statistical properties as robustness against the random attack because a targeting attack can be considered one instance of a random attack. The only difference between the two lies in the semantics of the chosen data portions, which is, from a statistical point of view, the same as any other instance of the same attack, hence the expected success of the detection process remains the same.

The $robustness$ value, or the "boundary attack strength", is used in step 4 to calculate the attacker's utility loss, for naïve and targeted attack scenarios. This means that the obtained utility loss is *at least* how much utility the attacker trades off for a successful attack.

We use the *German Credit* dataset for a detailed evaluation and discussion in Sections IV and V and additional two datasets, *Nursery* and *Adult Census*, to summarise our main findings in Section V-D. The datasets are available at the UCI Machine Learning repository [4] and described in Table II.

TABLE II
DATA SETS USED IN THE EVALUATION

Data set	#columns	#rows	target attribute	task
German Credit	20	1,000	good/bad credit risk	binary
Nursery	8	12,960	application rank	multi-class
Adult Census	14	48,842	income	binary

A. Horizontal subset attacks

The attacker's loss in the naïve scenario (Figure 3b) follows a steep trend of utility loss, by reaching $\sim$17% of a relative loss in model accuracy for the most robust scenario when 20 attributes and 100% of the rows marked. The bad utility overall is due to the high robustness of the scheme against the horizontal attack. From Figure 3a we can observe that even the mid-range parameter choice, such as $\{0.2, 20\}$ reaches robustness of 80%, meaning that in this case, the attacker needs to delete at least 80% of data rows to remove the fingerprint with a high probability. In the targeting attack, the overall attacker's loss decreases in the space of the most robust parameter settings, i.e. more than 20% marks Figure 3c. E.g. for settings that mark 20 attributes, the targeting attacker gets around 4% better utility results than the naïve. A similar trend applies to other settings. Even though the utility losses are not as high in comparison, they are still considerably high for the targeted attack. This means that the targeting attacker does improve their attacking attempts, however, the loss is still high enough to deter the attacker from applying any horizontal attack.

[4]http://archive.ics.uci.edu/ml

B. Vertical subset attacks

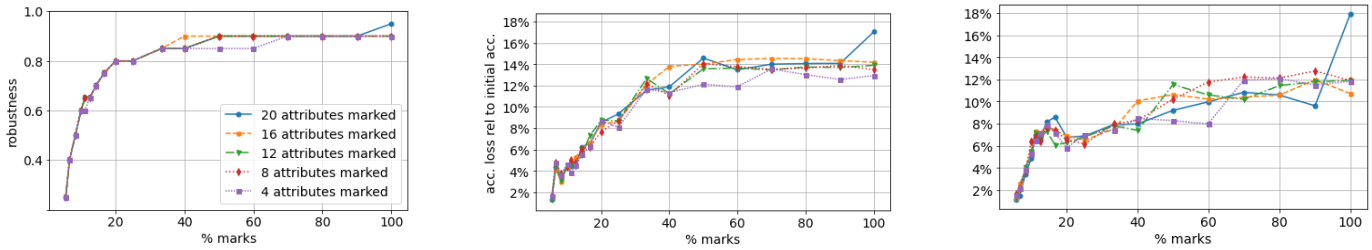
The feature selection method as a targeted vertical attack shows to be a very effective strategy for an attacker to shift their information loss toward less important features. The targeting attacker can considerably reduce the utility loss compared to the naïve attack, as shown in Figures 4b and 4c; in some cases the features selected by the attack even improve the use-oriented utility (i.e. have negative loss). When the attacker targets the column deletion, the effects of the fingerprint parameters are similar but less pronounced and stable compared to the naïve scenario. The robustness against the attack is still high for a good choice of parameters (Figure 4a). From these results we can read that the observed model has a relatively high effectiveness on a very small set of important features, and using more features does not improve the effectiveness significantly. If this is the case with the data set, then the attacker has the advantage of successfully removing the fingerprint while utilising a small subset of the features that yield similar performance as the original data. Because the feature selection can be a very successfully targeted attack, we introduce a countermeasure in the next section.

V. ADAPTIVE FINGERPRINT EMBEDDING

In this section, we discuss a pro-active defence against targeted attacks. To this end, the data holder may adapt the embedding pattern to anticipated threats, including both the naïve and targeted attack discussed in the previous section.

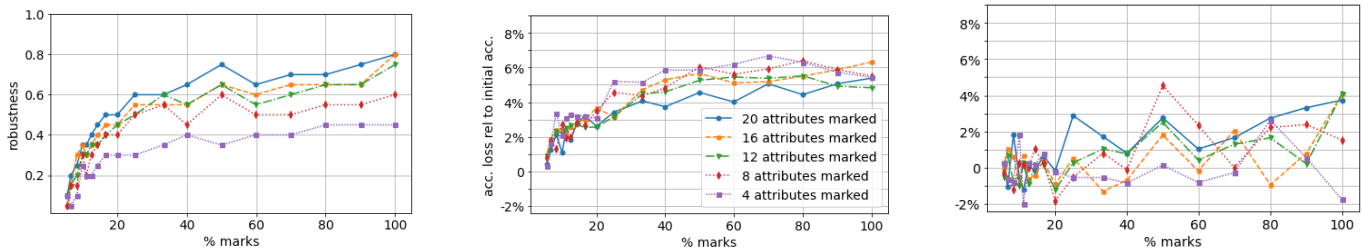
A. Fingerprint parameter choice

One way to think of the defence is to choose an appropriate parameter setting for the fingerprinting process that will result



(a) Robustness against the attack (b) Naïve attacker's utility loss (c) Targeting attacker's utility loss

Fig. 3. Horizontal subset attack on *German Credit* data. The utility is measured via performance of Gradient Boosting



(a) Robustness against the attack (b) Naïve attacker's utility loss (c) Targeting attacker's utility loss

Fig. 4. Vertical subset attack on *German Credit* data. The utility is measured via the performance of Gradient Boosting.

in a robust fingerprint. To enable this, three competing goals need to be met simultaneously:

1) Minimise the data utility loss caused by fingerprint
2) Maximise robustness of the scheme
3) Maximise attacker's utility loss

Since the fingerprint parameters have usually opposing effects on these goals (e.g. the number of fingerprint marks increases robustness, but decreases data utility), a good parameter setting achieves an acceptable trade-off between these. To exemplify

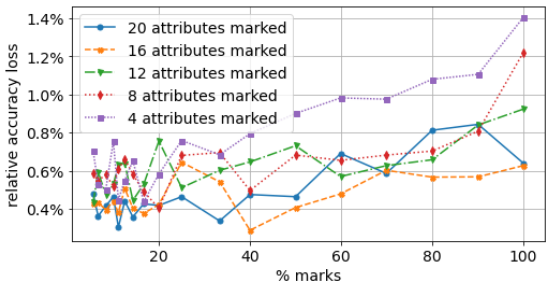


Fig. 5. Relative utility loss for gradient boosting due to fingerprinting *German Credit* data set

the process for selecting suitable parameters, we observe the robustness and utility results for *German Credit* data set. The most important choice is the value for parameter $\%marks$ since it affects the robustness and data utility the most, as seen in Figures 3, 4 and 6. For the values $\%marks > 0.6$, robustness against each of the attacks approaches and reaches its peak values. Since we need to minimise the $\%marks$ value for good utility, we can draw the lower boundary for robustness even lower, down to $\%marks = 0.3$, as the successful

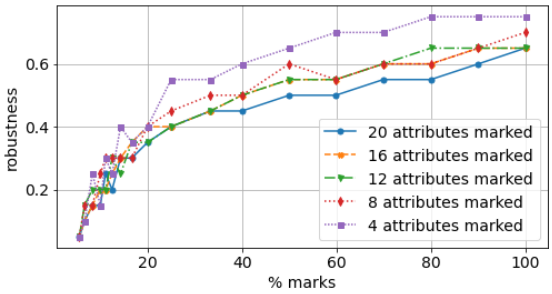


Fig. 6. Flipping attack on *German Credit* data

naïve attacks reduce data utility substantially for these values, hence it cannot be considered successful. Utility loss does not increase dramatically with higher $\%marks$ values as seen in Figure 5, so $\%marks = 0.3$ can be considered a good trade-off value.

The number of marked attributes affects the robustness against vertical and flipping attacks in opposing manners (cf. Figures 4a and 6). Thus, a compromise, intermediate value needs to be chosen, such as 80%. The results shown in Figures 3 to 6 are obtained when a random subsets of columns are chosen for fingerprinting. This leads the way for adaptive fingerprint embedding as a pro-active defence strategy, especially towards resisting targeted attacks, where, instead of random features, the defender can adopt a strategy for choosing them.

B. Adaptive selection of data columns for fingerprinting

We showed in Section IV-B that feature selection is a rather successful way of enhancing the vertical subset attack. We

discuss how the data holder (i.e. the defender) can pro-actively act, exploiting the fact that different attributes carry varying information values. Thus, instead of marking a random subset of columns, the defender can embed the fingerprint in a subset of columns chosen by some strategy (e.g. feature importance).

a) Defence access scenarios: This adaptive fingerprinting strategy requires defining new parameters for the fingerprinting process: the strategy for column selection and the size of the column subset. The white-box access, as defined thus far, would assume providing both the strategy and the number of marked columns accessible to the attacker. However, this scenario is trivial for the attacker because they have all the information necessary to perfectly remove the fingerprint from the data. Therefore, the *white-box targeting attacker* is defined under the following non-trivial attack scenario: the knowledge on the targeted selection of columns being utilised is known to the attacker, without the exact parametrisation (i.e. how many attributes are chosen and by which ranking scheme). If the knowledge accessible to the attacker is further restricted, the effectiveness of the adaptive defence is expected to differ (improve). To this end, we define the *grey-box targeting attacker* (re. Section III-A) for which the strategy of choosing the subset of marked columns stays disclosed. This way, it is harder for the attacker to target the marked columns, which is supported by our experimental results in Section V-C.

b) Attacker strategies: As a result of different levels of knowledge, the white-box and grey-box attackers have different attacking strategies. The best efforts of removing the fingerprint marks while keeping the acceptable utility for the grey-box attacker remains the targeted vertical subset attack as described in Section III-D1, whereas the white-box attacker can adapt their strategy using the additional knowledge. When the column selection strategy is known to the attacker, they can obtain the exact feature importance values as the defender. Although the exact number of marked columns is not known, a better strategy than removing the least important features is to remove the features from the mid-range – in other words, removing a feature of medium importance is a good choice because it might be among those chosen by the defender, yet does not introduce as much utility degradation as removing the most important features. Such a strategy will be evaluated via the worst-case scenario for the defender: the case where the attacker guesses the exact number of marked features and removes a portion (or all) of the least important, marked features.

C. Evaluation

For the evaluation of the adaptive fingerprinting strategy, we follow the recommendations presented in Figure 2, and thus evaluate the effects on data utility and the effectiveness of one instance of the adaptive defence strategy on *German Credit* data. Our defender chooses the attributes for marking based on the mutual-information feature selection strategy [22]. Firstly, the adaptive strategy should show effectiveness in thwarting the vertical attack while not introducing significant utility losses to the data, hence we evaluate the effects on accuracy of the best performing model for the task, i.e. gradient boosting. Secondly, the robustness against the adaptive vertical attack is evaluated against two targeting attackers, white-box and grey-box, to show the said effectiveness. Lastly, we need to ensure that the adaptive strategy does not gain its effectiveness against targeting attacks at the cost of reduced effectiveness against naïve attacks, hence the robustness is as well evaluated against the naïve white-box attacker.

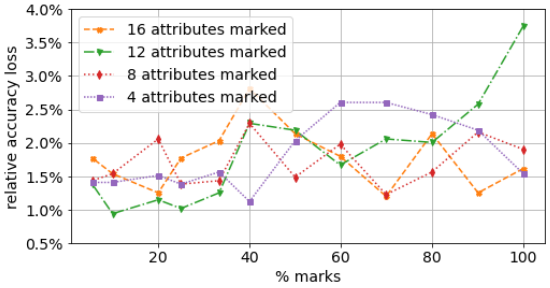
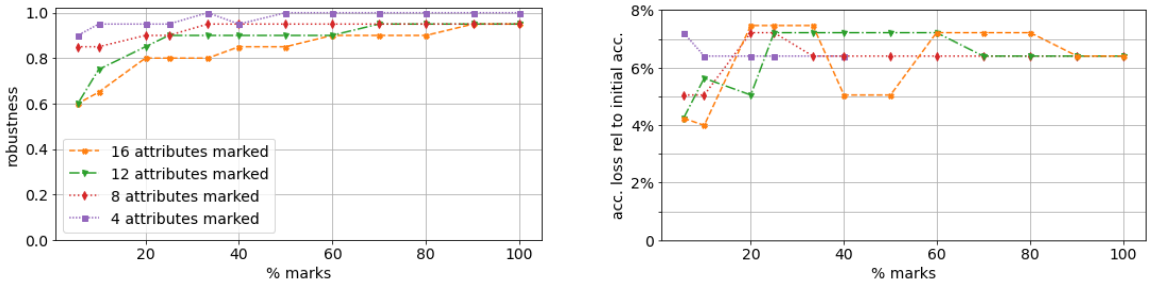


Fig. 7. Utility loss after applying adaptive fingerprinting

a) Utility loss of adaptive fingerprinting: The differences in utility loss between the untargeted (Figure 5) and adaptive fingerprint embedding (Figure 7) suggest that the defender gives up a only minor amount of measured utility in an attempt to raise robustness against the targeted attacks – in most cases only around 1% of the relative accuracy loss.

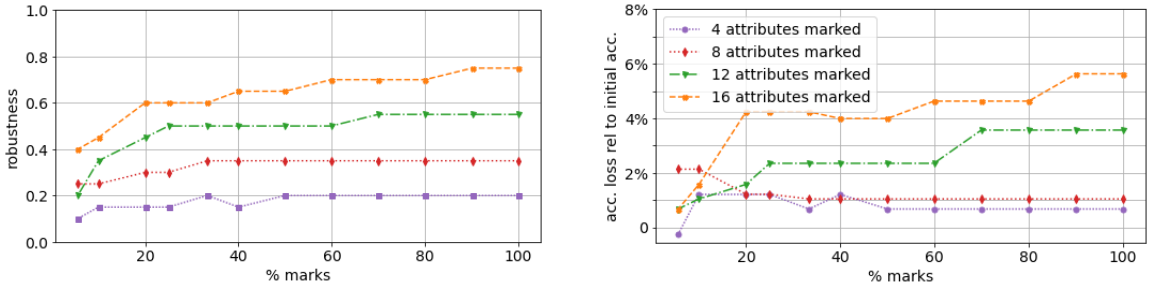
b) Grey-box setting: The gain of robustness against a targeted vertical subset attack is indeed large when using the adaptive fingerprint embedding in the grey-box setting, as shown in Figure 8a, compared to the robustness of a standard approach against the same attack (Figure 4a). The robustness overall increases by around 0.2, for most of the parameter settings. In line with better robustness, the attacker's loss also gets larger due to having to apply stronger attacks to successfully remove the fingerprint, as seen by comparing Figures 4c and 8b. These losses are caused by the fact that the attacker needs to delete a large number of columns that are important for the predictive tasks, as deleting the lesser important (and thus not fingerprinted) columns does not help in removing the fingerprint. This showcases the effectiveness of the adaptive defence strategy.

c) White-box setting: The white-box attacker, however, has the advantage of knowing the feature selection strategy of the defender and can adapt their attack strategy accordingly. We here evaluate whether the defence can persist under the disclosed strategy of choice of columns for marking. Figure 9 shows the worst-case scenario when all of the columns deleted by the attacker are indeed those that are fingerprinted. For this reason, the robustness in Figure 9a shows inverted behaviour depending on the number of marked attributes compared to Figure 8a because the chance of the attacker "missing" the marked column is excluded from these observations. The robustness hence decreases significantly for a low amount of marked columns but persists on a similar level for 12 or 16 marked attributes (60% and 80% of the data columns,



(a) Robustness (b) Attacker's utility loss

Fig. 8. GREY-BOX: Adaptive defence vs. targeted vertical subset attack on *German Credit* data set



(a) Robustness (b) Attacker's utility loss

Fig. 9. WHITE-BOX: Adaptive defence vs. targeted vertical subset attack on *German Credit* data set

respectively). The attacker's utility in Figure 9b decreases significantly, as well, compared to the grey-box setting in Figure 8b. For the most robust evaluated scenario of marking 16 attributes, the utility decrease of 4-5% can be enough to deter the attacker from attacking.

It is important to note that this is the worst-case scenario for the defender, when, by chance, all of the columns deleted by the attacker are marked. In reality, this is not expected to be the case because the number of marked columns is still undisclosed in the white-box setting.

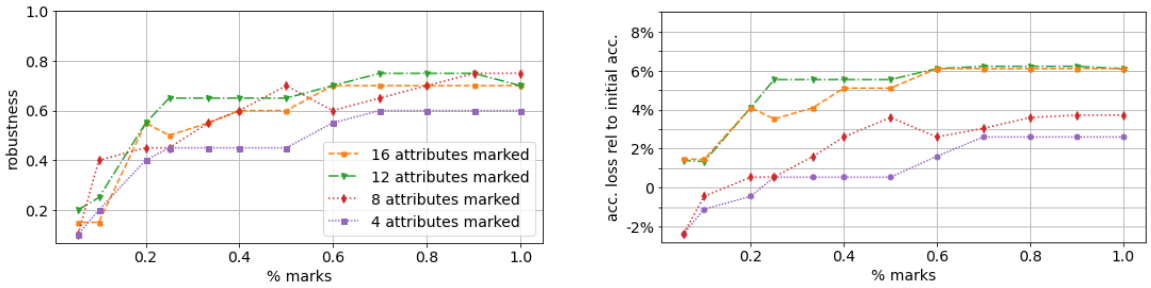
d) Adaptive defence under naïve attacks: Although the adaptive fingerprinting is successful against targeted attacks, we need to ensure that it does not compromise the robustness against a naïve attacker. Comparing the results in Figure 10 to the results of the standard embedding against a naïve attacker in Figure 4, we can see very similar trends in robustness and attacker's utility loss. This confirms that the adaptive fingerprint does not degrade its robustness against a naïve attacker – the naïve attacker has roughly equal chances of deleting the marked columns regardless of the defender's strategy since the attacks are applied randomly.

D. Discussion

The effectiveness of 3 fingerprinting strategies (classical, grey box and white box) is summarised in Table III, where it is expressed in terms of robustness, the utility cost of achieving that robustness and the amount of utility the attacker can gain by targeting the attack instead of attacking randomly (naïve). The results are obtained for parameters with a good robustness-utility loss trade-off for each of the 3 data sets. In the ideal, white-box access scenario, the adaptive fingerprinting is a robust solution with a relatively low impact on the utility, however, this is dictated by the dataset properties as we can see that the approach did not succeed for the Nursery dataset (probably due to the high discrepancies between the most important feature and the rest). In these cases, the grey-box scenario is a good alternative since the robustness reaches 100% with low utility cost. The downside of the grey-box scenario is that the information about the marked attributes needs to be securely stored for the detection process.

TABLE III
SUMMARY OF THE EFFECTIVENESS OF FINGERPRINTING STRATEGIES: CLASSICAL, GREY BOX AND WHITE BOX

Fingerprinting approach:	classical	grey box	white box
German C. {%attr,%marks}	{100%,100%}	{20%,100%}	{80%,100%}
robustness	80%	100%	75%
utility loss	0.64%	1.51%	1.51%
attacker gains by targeting	4.68%	-	0.50%
Nursery {%attr,%marks}	{100%,100%}	{40%,25%}	{80%,60%}
robustness	75%	100%	75%
utility loss	6.54%	1.74%	7.36%
attacker gains by targeting	51.02%	-	52.40%
Adult {%attr,%marks}	{100%,100%}	{50%,25%}	{80%,50%}
robustness	86%	100%	85%
utility loss	0.08%	0.38%	0.46%
attacker gains by targeting	4.94%	-	0.75%



(a) Robustness

(b) Naïve attacker's utility loss

Fig. 10. "*Regression testing*": Adaptive defence vs. naïve vertical subset attack on *German Credit* data set

VI. Conclusions and Future Research

This paper summarises important aspects of creating a high-quality fingerprint: low incurred utility loss, and high robustness, in conjunction with a large utility loss for the attacker as a side-effect of an attempt to remove the fingerprint. We also showed the impact of fingerprint parameters on these three key aspects. Furthermore, we extend the attacker model against fingerprinting relational data by defining a stronger version of the attacker: a targeting attacker that uses background knowledge about the data and its intended downstream purpose to increase their chances of a successful attack, i.e. removing the fingerprint while at the same time compromising the utility of the data with the attack to lesser extents. Some flavours of the targeting attacker are very successful, especially the adaptive vertical subset attack, where the attacker removes columns that are of low importance for the assumed predictive task for the data. In answer to the targeted attacks, we propose an adaptive fingerprint embedding, as a pro-active defence. This type of defence anticipates the potential threats of targeting attackers and modifies the fingerprint embedding process accordingly. We show that our most successful novel targeted attack can be countered by embedding the fingerprint into the most relevant attributes of the dataset. With a low cost of utility loss due to this targeted fingerprint, such embedding makes it practically impossible for the targeting attacker to successfully remove the fingerprint. At the same time, the targeted fingerprint does not affect the robustness against naïve attacks, i.e. it does not expose a novel risk.

In future work, we will extend the attacker model with more types of targeted attacks, e.g. a targeted superset attack and consequently, focus on further adaptions of fingerprinting to resist these attacks. A further focus will be on guidelines for selecting fitting parameters for robust and utility-preserving fingerprinting, especially on finding the trends that would lead to (semi-)automating the selection.

References

[1] R. Halder, S. Pal, and A. Cortesi, "Watermarking techniques for relational databases: Survey, classification and comparison." *J. Univers. Comput. Sci.*, vol. 16, no. 21, 2010.

[2] M. Kamran and M. Farooq, "A Comprehensive Survey of Watermarking Relational Databases Research," *arXiv:1801.08271*, 2018.

[3] W. Trappe, M. Wu, Z. Wang, and K. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, 2003.

[4] M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik, "Natural language watermarking: Design, analysis, and a proof-of-concept implementation," in *International Workshop on Information Hiding*. Springer, 2001.

[5] R. Venkatesan, V. Vazirani, and S. Sinha, "A graph theoretic approach to software watermarking," in *International Workshop on Information Hiding*. Springer, 2001.

[6] X. Zhao, Q. Liu, H. Zheng, and B. Y. Zhao, "Towards graph watermarks," in *ACM Conference on Online Social Networks*, 2015.

[7] E. Yilmaz and E. Ayday, "Collusion-resilient probabilistic fingerprinting scheme for correlated data," *arXiv:2001.09555*, 2020.

[8] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: schemes and specialties," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, Jan. 2005.

[9] J. Lafaye, D. Gross-Amblard, C. Constantin, and M. Guerrouani, "Watermill: An optimized fingerprinting system for databases under constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, 2008.

[10] E. Al Solami, M. Kamran, M. Saeed Alkatheiri, F. Rafiq, and A. S. Alghamdi, "Fingerprinting of relational databases for stopping the data theft," *Electronics*, vol. 9, no. 7, 2020.

[11] S. Liu, S. Wang, R. H. Deng, and W. Shao, "A Block Oriented Fingerprinting Scheme in Relational Database," in *Information Security and Cryptology (ICISC)*. Berlin, Heidelberg: Springer, 2005.

[12] F. Guo, J. Wang, and D. Li, "Fingerprinting Relational Databases," in *ACM Symposium on Applied Computing (SAC)*. ACM, 2006.

[13] R. Sion, "Proving ownership over categorical data," in *Proceedings. 20th International Conference on Data Engineering*. IEEE, 2004.

[14] E. Bertino, B. C. Ooi, Y. Yang, and R. H. Deng, "Privacy and ownership preserving of outsourced medical data," in *International Conference on Data Engineering (ICDE)*. IEEE, 2005.

[15] T. Šarčević and R. Mayer, "A correlation-preserving fingerprinting technique for categorical data in relational databases," in *IFIP Int. Conf. on ICT Systems Security and Privacy Protection*. Springer, 2020.

[16] R. Agrawal and J. Kiernan, "Watermarking relational databases," in *Int. Conference on Very Large Databases (VLDB)*. Elsevier, 2002.

[17] T. Šarčević and R. Mayer, "An evaluation on robustness and utility of fingerprinting schemes," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2019.

[18] T. Ji, E. Yilmaz, E. Ayday, and P. Li, "The curse of correlations for robust fingerprinting of relational databases," *arXiv:2103.06438*, 2021.

[19] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 2015.

[20] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018.

[21] I. Mani and I. Zhang, "kNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction," in *Workshop on Learning from Imbalanced Datasets*. ICML, 2003.

[22] A. Kraskov, H. Stögbauer, and P. Grassberger, "Erratum: Estimating mtual information," *Physical Review E*, vol. 83, no. 1, 2011.