Will Sharing Metadata Leak Privacy?

Danning Zhan
Web Information System
TU Delft

Rihan Hai
Web Information System
TU Delft

***Abstract*—In the dynamic field of data management and machine learning, achieving a balance between effective data use and privacy preservation is increasingly crucial. Federated learning exemplifies this challenge by training machine learning models on data distributed across isolated silos while adhering to privacy regulations like GDPR. A key aspect of this process involves sharing metadata, such as feature names, essential for model accuracy. Yet, the privacy implications of this metadata exchange have been largely unexplored.**

This paper examines the potential privacy risks of communicating detailed metadata in federated learning frameworks. While metadata is critical for enhancing data utility and supporting advanced analytics, we address the paradox that it might inadvertently lead to privacy violations. We focus on functional dependencies (FDs) and relaxed functional dependencies (RFDs), which are crucial metadata types in database design and data quality. We aim to define data privacy formally and investigate how sharing these dependencies affects privacy preservation, using probabilistic methods and analytical discussions to understand their impact.

***Index Terms*—Privacy, Relaxed Functional Dependencies**

I. Introduction

The accumulation of data has heightened data privacy concerns, prompting stricter enforcement of regulations such as GDPR [24], HIPAA [5], IPA [2], and PIPL [3]. Data collectors are mandated to ensure the confidentiality of their amassed data, leading to the formation of data silos. Despite the potential insights from cross-silo data processing, regulatory measures to safeguard data privacy hinder such collaborations. Federated learning emerges as a viable data processing approach that upholds privacy standards amidst these constraints.

Federated learning involves training ML models using data residing in isolated silos while preserving data privacy. According to how the feature space and sample space are partitioned among the data sources, federated learning can be categorized as vertical federated learning (VFL) [15] and horizontal federated learning (HFL) [26]. For VFL, data silos share overlapping data instances but disjointed attributes, whereas for HFL, data silos share overlapping attributes with different data instances. HFL typically operates under the same or similar database schema among participants. This paper focuses on the more complex scenario of VFL, where participants with differing database schemas must exchange metadata before model training.

Fig. 1 shows a fintech scenario involving two distinct parties: party A is a bank, and party B is an e-commerce company. Each has accumulated distinct datasets about a common

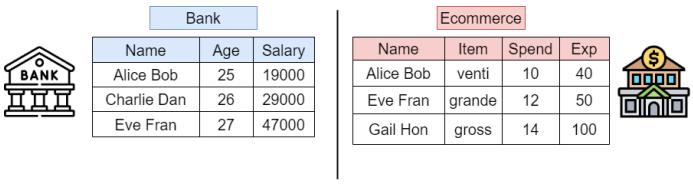

Fig. 1. Vertical Federated Learning between a bank and a e-commerce company.

population. This presupposes that the dataset originates from a homogeneous population that is known. Recognizing the value that data heterogeneity can bring to their analysis, both entities might consider engaging in vertical federated learning to process their data collaboratively. An essential preliminary step in this process involves metadata exchange between the parties to enhance mutual understanding of data, which is vital to ensuring the utility of any downstream models.

Metadata, covering information about a dataset, such as attribute name, range, and type, as well as constraints between attributes, such as functional dependency (FD) and relaxed functional dependencies (RFD) [6], [9], [13], [18], play a crucial role in data processing tasks such as data cleaning [7] and data integration [25]. Yet, the privacy implications of metadata exchange, particularly functional and relaxed dependencies, remain unexplored in VFL literature. This study investigates the privacy impacts of sharing these dependencies, comparing them to the situation where this information is withheld. We concentrate on VFL due to the predominance of FDs and RFDs identified within the data. VFL exhibits distinct metadata dependencies among participants, contrasting with HFL, where metadata are similar. Our analysis will leverage communicated metadata to assess data privacy implications quantitatively using probabilistic methods.

Contributions Our contributions are outlined as follows.

- We will provide a *formal* definition of privacy leakage in the context of vertical federated learning.
- Using probabilistic methods, we will examine if sharing various metadata types, including attribute names, domains, functional dependencies, and relaxed functional dependencies, could result in privacy leakage.

II. Preliminaries

In this section, we will formally define concepts that we will use throughout this work.

TABLE I
NOTATION DEFINITIONS

Notation	Definition
R_{real}	The real relations
R_{syn}	Generated/Synthesized relations
$A \subset R$	Subset of attributes of the relations
$a \in A$	Any value a in the domain of attribute A
t_i, r_i, u_i	Tuple at index i, index being the row index
$t_i[A]$	Attribute A associated with the tuple i
d()	Any valid metric (distance) function
θ	Probability value
E()	Expected value of random variable
D_y , Dom(Y)	Domain of attribute Y
$\|.\|$	Cardinality or Size of the set

TABLE II
EXAMPLES TABLE: EMPLOYEE

Name	Age	Department	Salary
Alice	18	Sales	20000
Bob	22	Customer Service	25000
Charlie	22	Sales	27000
Danny	26	Management	35000

A. Functional Dependencies

Functional dependencies (FDs) are one of the most important constraints in relational database design, e.g., normalization [19]. A functional dependency $X \rightarrow Y$ specifies that attributes Y depend on attributes X. We use the following definition of functional dependency on relation R of party A in Figure 1 [13]. If we consider the attributes $X, Y \subset R$ such that $X \rightarrow Y$ is a functional dependency if and only if $\forall$ tuples $t, r \in R, t[X] = r[X] \implies t[Y] = r[Y]$. This means that for any tuple in the relation of party A, the attribute X of party A will allow us to determine the value for attribute Y. We can define the same definitions for party B's data if we consider the data scenario of VFL in Figure 1.

Relaxed functional dependencies (RFDs) hold under less constrained conditions than functional dependencies [9]. Because of the less-constrained definition of the dependencies, this would imply that more functions and conditions can be utilized for RFDs.

Example 2.1: TABLE II is an example table of employees with four attributes: name, age, department, and salary. In the example, we assume the attribute 'name' is unique. Two possible functional dependencies are $Name \rightarrow Age$ and $Name \rightarrow Salary$. The table also shows relaxed functional dependencies $Name \rightarrow Salary$ and $Age \rightarrow Salary$.

B. Privacy Leakage

In the context of vertical federated learning, employing an adjusted definition of privacy leakage that aligns more closely with regulatory standards, particularly those outlined in the GDPR [1], [5] is imperative. To effectively address this requirement, we propose establishing a quantitatively measurable definition of privacy leakage. This definition aims to accurately assess the extent of data privacy breaches or exposures that may occur within the framework of vertical federated learning, ensuring compliance with GDPR mandates. This approach adheres to regulatory prerequisites and provides a standardized metric for evaluating privacy risks in a federated learning environment.

Participating parties exchange dataset-related information in the preliminary stage of model training within a vertical federated learning framework, specifically metadata that describes the content of their respective data. This metadata exchange, while necessary, poses a potential risk for privacy leakage. Consider a scenario involving two parties, A and B. When party A communicates its metadata with party B, there arises a possibility that party B might use this metadata to construct a synthetic dataset, essentially an inferred approximation of A's real dataset. We denote A's real dataset as R_{real} and the synthetic dataset generated by B as R_{syn}. Privacy leakage is a concern if we can identify data points or values in R_{syn} that closely match or replicate those in R_{real}.

To understand and measure the privacy risk, we define '*identifiability*' – a fundamental concept in the GDPR[1].

Definition 2.1: (**Identifiability**) We say a tuple t is identifiable if, for the relation R_{real} of the real data, $\exists A \subset R_{real}$ such that for any value $a \in A$, $\exists! t \in R$ such that $t[A] = a$.

The above definition of identifiability hinges on whether a unique data point can be isolated based on a specific set of values for a given set of attributes A. If such a set exists where only one data point equals the tuple value of A, the data point is classified as identifiable. Thus, anonymization techniques [11] aims to ensure that shared data remain non-identifiable, upholding privacy standards and aligning with data protection regulations like the GDPR.

Identifiability pertains to the ability to link specific data elements within a dataset (in this case, R_{syn}) to their counterparts in another dataset (R_{real}) during data sharing. Next, we discuss privacy in the context of vertical federated learning, which has a different characteristic than data sharing.

Privacy for VFL. Federated learning aims to train machine learning models across multiple data sources, or 'silos', while preserving privacy. Before the training begins, data from various parties is synchronized using private set intersection techniques [10], [12]. This process ensures that the identity of the data tuples is known only to the parties involved in the training. This unique aspect of Vertical Federated Learning calls for a more nuanced understanding of privacy leakage. To address this, we propose a distinct definition of privacy leakage for VFL. Since categorical and continuous data types have different impacts on privacy risks, we will define the privacy leakage for each data type separately.

Definition 2.2: (Categorical data privacy leakage for VFL) Consider the scenario where R_{real} and R_{syn} represent the relations in the real dataset and synthetic dataset, respectively. Let A be a subset of categorical attributes in R_{real}. For any given tuple t_i, which exists at the same index i in both R_{real} and R_{syn}, we observe a privacy leakage if the attribute

[1]https://gdpr-info.eu/art-5-gdpr/

values of t_i in A are identical in both R_{syn} and R_{real}, i.e., $t_i^{\text{syn}}[A] = t_i^{\text{real}}[A]$.

The proposed definition of privacy leakage considers the correspondence between the i^{th} tuple in the real and the synthetic dataset. Privacy leakage occurs if values match within these tuples, as distinct values for categorical attributes carry divergent meanings. The index assumes critical importance in the context of VFL due to its source being the intersection of collaborating entity's datasets.

For continuous attributes, a more nuanced definition of data privacy is required. Given the extensive variability inherent in continuous variables, theoretically, they possess infinite possible exact values. Consequently, if a generated value falls within a parameterized neighborhood of the real attribute value, measured using a specified distance metric $d()$ such as Euclidean distance [8], it should be classified as a privacy leakage, which leads to the below definition.

Definition 2.3 (Continuous Data Privacy Leakage in VFL): Consider the scenario where R_{real} and R_{syn} represent the relations in the real dataset and synthetic dataset, respectively. Let A be a subset of continuous attributes in R_{real}, and ϵ be a given error threshold. For any given tuple t_i, which exists at the same index i in both R_{real} and R_{syn}, we observe a privacy leakage if the distance d between $t_i^{\text{syn}}[A]$ and $t_i^{\text{real}}[A]$ is within the error threshold, i.e., $d(t_i^{\text{syn}}[A], t_i^{\text{real}}[A]) \leq \epsilon$.

III. Privacy Analysis of Metadata

Metadata describes the data, for example, attributes and domains, and outlines the table's dimensions. FDs and RFDs indicate the dependencies among attribute values and inherent data structures. In current federated learning frameworks [12], [16], metadata such as attribute names and domains are commonly exchanged among different parties. However, the potential privacy implications of sharing this expanded metadata scope remain unexplored.

A critical concern is that sharing metadata can jeopardize data privacy. This risk stems from having access to metadata, allowing for data generation based on the underlying data structures. This process of data generation, informed by the knowledge of metadata, poses a significant threat to data privacy. Therefore, our study aims to examine whether sharing various types of metadata, including attribute names and domains (Section III-A), functional dependencies (Section III-B), and relaxed functional dependencies (Section IV), might inadvertently lead to privacy breaches.

A. Attribute Name and Domain

We first discuss whether sharing attribute (feature) names and domains will leak privacy.

Privacy Analysis. When generating a synthetic dataset, the generation of each tuple is independent; this implies that generating the entire data set will follow a binomial distribution. Given attribute A, we denote θ_A as the probability of generating the actual value of A correctly, i.e., the same as the real dataset. Letting $D_A = Dom(A)$, we can see that for random generation from a uniform distribution, $\theta_A = \frac{1}{|D_A|}$. The binomial distribution will give us the expected number of correct generations as $N\theta_A$, where N is the total number of data instances. As N, and θ_A are non-zero, a positive number of values can be generated correctly, so following Definition 2.2, if $N\theta_A \geq 1$, we will have privacy leakage.

Example 3.1: Considering the attributes in Table II and the attribute Age and department, which has a domain from [18,26] containing 9 values. So then the probability of generating a tuple with the identical age value as the real dataset is $\frac{1}{9}$. Given there are only 4 data points, there will be a probability of $\frac{4}{9}$ of generating any age correctly within the table, so the likelihood of privacy leakage is low. However, for the domain of department, there are only 3 departments, so the probability of random generating the department is $\frac{1}{3}$, so the expected value would be $\frac{4}{3}$, meaning that we would expect there to be one correct guess.

B. Functional Dependency

Consider a functional dependency from attribute $A \rightarrow B$, having of domains $D_A = Dom(A), D_B = Dom(B)$ respectively; we can infer the probability of correct generation to be $\theta_A = \frac{1}{|D_A|}$ and $\theta_B = \frac{1}{|D_B|}$. The expected value of generating the correct value of A will be $N\theta_A$.

Privacy Analysis. Given a functional dependency $A \rightarrow B$ and $a_i \in D_A$, $B = \bigcup \pi_A$. For generating the mappings that satisfy the dependencies $A \rightarrow B$, the probability of generating the mapping from A to B is $P(B|A = a_i) = \frac{1}{D_B}$. Data generation occurs once across the dataset, which are determined by functional dependencies. An FD $A \rightarrow B$ reveals underlying relationships A and B, allowing for their one-time initialization throughout the dataset without necessitating explicit knowledge of each dependency, thereby facilitating the derivation of relationships directly from their definitions.

The FD $A \rightarrow B$ indicates that A refines B, meaning that $|D_A| \geq |D_B|$. Following a binomial distribution, the number of correctly generated dependencies is $E(B|A) = \frac{|D_A|}{|D_B|} \implies E(B|A) \geq 1$. It means there will be at least one correct mapping between A and B. However, the total expected value of generating the correct values of A and B is $N\theta_A \frac{E(B|A)}{|A|}$, which would be the same as random generation. The generation of the values of attributes A, B will solely depend on the values of the synthetic dataset; we have the expected value of $\frac{n}{|D_A|}$ values are correct. Then, for all of the correct values, we assume the number of correct values will be dependent on the domain of A.

Example 3.2: Consider the example in Table II, and the FD $Salary \rightarrow Age$. Assuming the domain of Age is in the range of values [18, 26] and the salary is in the range $[20, 35]$ in thousands; there is an FD between salary and age. We can generate values for salary such as $\{20, 25, 30, 35\}$, and then the mappings can be generated as $20 \rightarrow 20$, $25 \rightarrow 23$, $25 \rightarrow 25$, $35 \rightarrow 26$. This mapping satisfies the definition of functional dependency. From the mapping, if tuples have a salary of $20k$, then the age value is 20.

A correct mapping would mean the mapping is always correct; consequently, if the mapping is incorrect, it would

always be incorrect. Whereas, for random generation, correctly generating the age attribute will not depend on the generation of the salary attribute.

The above argument can be extended to more than one functional dependency. Consider 3 attributes $A, B, C \subset R$, of relation R , with domain $D_A = Dom(A), D_B = Dom(B), D_C = Dom(C)$. The property of transitivity, states that if $A \rightarrow B$ and $B \rightarrow C$, then the value of A will decide B, which in turn decides the value of C. The mapping generation between B and C is an identical procedure to that utilized for A and B, given the independence of these dependencies. Relative to the current information disclosure level, communicating functional dependencies, in conjunction with attribute names and domains, does not exacerbate privacy leakage.

IV. Privacy Analysis of RFDs

Extensive research has been conducted on relaxed functional dependencies (RFDs), with 35 prominent types of RFDs compared in a survey [9]. However, a significant proportion of these RFDs represent variations of a core set. Therefore, in this section, we have selectively conducted a privacy analysis of a subset of representative RFDs, with the arguments applicable to their variations.

A. Approximate Functional Dependency

Approximate functional dependency (AFD) is a common variation of regular function dependency with the inclusion of the g_3 error [14]. The g_3 error states that given relations R, there is a subset of relations $\exists R_1 \in R$ such that for attributes $A, B \subset R$ the functional dependency $A \rightarrow B$ holds on the relation R without excluding R_1. The error term is defined to be $\epsilon = \frac{|R|-|R_1|}{|R|}$, meaning that for this RFD, if an ϵ proportion of points are removed from relation R then the functional dependency would hold. This can be seen to be similar to the probabilistic functional dependency (PFD) definition, except being defined on the entire schema, it is defined on the partitions of A [21].

Privacy Analysis. The expected value will include a factor for $\frac{1}{\epsilon}$ to the expected value of strict functional dependency. Implying that we will have an $\frac{N\epsilon}{|A||B|}$ of random generated data and $\frac{N\epsilon}{|A||B|}$. The total amount of correct data generated will be the same as strict functional dependency and random generation. Similar to functional dependency, the $1-\epsilon$ proportion will belong to a subset of partitions of A, and the ϵ proportion of data generated correctly will be scattered across all partitions. We will arrive at the same expression for FD, so the privacy conclusion for AFD is the same as FD.

B. Numerical Dependency

Numerical Dependency enforces a cardinality constraint on attribute X, stipulating that each value of X is associated with no more than K values in attribute Y. This constraint means that given the domain of Y, there are at most K unique values that each value of X can map to.

Privacy Analysis. For each attribute of X, we would have the probability $\theta_X = \frac{1}{|X|}$ of correctness leading to the expected value of $N\theta_X$, and generating mapping from $X \rightarrow Y$ follows a hyper-geometric distribution [20]. Generation of the mapping from X to Y would depend on random selection, or we can assume that the mappings all have the same cardinality. As for each value of X, the domain of attribute Y is partitioned into one that is valid in the original and one that is not. Through random selection, the expected value of correct mappings is $n\frac{K}{N}$, where $N = |D_Y|$, $D_Y = Dom(Y)$ and K is the cardinality of the partition of X, n is the number of samples we are selecting.

For the generation of attribute Y, the mappings generated that would be correct would be $k\frac{K}{N}$, where the assumption is that $N >> K$. For each value of X, by the hyper-geometric distribution, the expected value of generating correct mappings is $\frac{k^2}{|D_Y|}$. The probability of finding at least one correct mapping that satisfies the dependency is thus $1 - \frac{\binom{(|D_Y|-K)}{K}}{\binom{|D_Y|}{K}}$. The probability of selecting the correct map from the generated mappings is selecting $\frac{K}{|D_Y|}$. As the selection is independent, the probability of selecting X and Y values both correctly has a probability of $\frac{K}{|D_Y||D_X|}$. This means that the expected value of generating the pair of attributes X and Y both correctly is $\frac{NK}{|D_X||D_Y|}$. However, there is the possibility that that is not the case, that if $k > \frac{N}{2}$, then there would be guaranteed at least $|\frac{N}{2} - k|$ dependencies correct. This situation would significantly increase the expected value for privacy leakage.

C. Order Dependency

Unlike approximate and functional dependencies predicated on equality, order dependency is established based on inequality and ordering. This applies to categorical and continuous attributes, as ordering is definable across all variable types. We can define the dependency to be for a relation R if we let $X, Y \subset R$, such that $\forall t, u \in R$ if $t[X] \leq u[X] \rightarrow t[Y] \leq u[Y]$. If there is an order dependency from attribute X to Y, for any tuple, if X increases, attribute Y's value will increase, i.e., the ordering attribute X will also order attribute Y. Suppose we consider the order dependency from categorical X to continuous Y attributes. In that case, this implies that we know the number of intervals, as the definition of the order dependency gives this. Thus, we can create a sequence of $y_i, i \in 1, ...n$ where n is the size of the domain of X. Forming partitions of the domain of Y into intervals denoted by $[y_i, y_{i+1}]$.

The alternative would be if we have the order dependency defined between continuous attributes X and Y, we can similarly discretize the domain of X. We can create intervals within the domain of Y that satisfy order dependency. The discussion for categorical attributes is discussed in Section IV-E.

Privacy Analysis. Suppose there are two sequences, $\{y_i^{'}\}$ and $\{y_i\}$ over the same domain. We want to obtain the probability of the intersections of the intervals created by the sequence $[y_i^{'}, y_{i+1}^{'}]$ over the domain of the attribute Y. For the first interval we re generating $y_1^{'}$ against y_1. The probability that we generate within the right interval is $\frac{y_1^{'}-y_1}{|D_Y|}$. This probability

of correctness will be given by $\frac{\text{overlap of the interval}}{\text{total remaining size of the domain}}$. We are generating conditions on the previous interval for the next and hereafter intervals. For generating $y_2^{'}$ we know that it will be greater than $y_1^{'}$, so we must consider what will happen if we are generating $y_2^{'} < y_1$ and when $y_1^{'} > y_2$, then the overlap of the intervals $[y_1^{'}, y_2^{'}]$ and $[y_1, y_2]$ is $\emptyset$. If the overlap is the empty set, this would imply a zero probability; otherwise, the probability would be non-zero. The probability that the value for attribute Y will be generated given value x_i becomes $\theta_{y_i} = \frac{max(y_{i+1} - y_i^{'}, 0)}{y_{max} - y_i}$.

This probability will be used for each of the intervals to find the probability is $\theta_{x_i} * \theta_{y_i}$ overlap between the initial and terminal intervals due to a fixed domain. Consequently, the probability that at least two partitions of attribute A will exhibit non-zero values is guaranteed. The expected value for the number of data instances that we will be able to generate is given by two binomial distributions, with $N\theta_X\theta_Y$ number of points generated correctly. The total expected value will be $\sum_i N\theta_{x_i}\theta_{y_i}$, where θ_{y_i} is defined as above.

Because of the high variance in this dependency defined for continuous features, we would expect the error to be quite high implying that the privacy leakage would be quite low. It would also depend very heavily on the partitioning of the domain of Y depends very heavily on the distribution of Y. However, this distribution is not communicated, so we will assume a uniform distribution for our experiments.

D. Differential Dependencies

These dependencies arise from the principle of differentiation, where, given a metric for attribute X, a corresponding dependency is established for a metric defined on attributes Y. [22]. As we are working with continuous variables, we will consider that having an error of ϵ would still leak privacy. So we would have the expected value of $n\frac{2\epsilon}{|D_X|}$ of correctly generated points, where $D_X = Dom(X)$.

To fulfill this requirement, it is necessary that for a given metric applied to attributes X, the attributes Y must meet their corresponding metric criteria. This implies that if values of X in tuples are proximal, then the Y values in those tuples should also be proximal. To uphold this principle, intervals within the domain of Y can be established to mirror the intervals on the domain of X. In other words, given relation R, we have $\forall t \in R$ such that $\forall \epsilon > 0, \exists \delta; \forall t[X] \in [x_i - \epsilon, x_i + \epsilon] \implies t[Y] \in [y_i - \delta, y_i + \delta]$.

Privacy Discussion of Differential Dependency. For every value of X, a ball can be constructed around it, ensuring that any value within this ball is mapped to a value within a corresponding ball centered around the Y attribute's value. A ball is a set of values of the attribute that are within an equal distance from a specific value. We generate the correct value with respect to the interval $[x - \epsilon, x + \epsilon]$, this has a probability of $\frac{2\epsilon}{|D_X|}$. Similarly, the probability of generating the correct Y would follow a Markov process [17], as the interval would be defined relative to previous intervals. If a value resides within an interval previously generated, then the intervals of attribute X will exhibit overlap, leading to a consequent overlap in the intervals of attribute Y. The likelihood of accurately generating the overall mappings is determined by the product of overlaps, normalized by the range. The generation of the attribute X will have a probability of $\frac{2\epsilon_x}{range(x)}$. For the attribute Y there is a probability of $y^{'} \in R_{syn}$, $y \in R_{real}$ the probability of generating the correct Y attribute value is $\frac{[y^{'} - \epsilon, y^{'} + \epsilon] \cap [y - \epsilon, y + \epsilon]}{range(Y)}$. Giving us an overall probability of generating the correct value to be $\frac{2\epsilon_x [y^{'} - \epsilon, y^{'} + \epsilon] \cap [y - \epsilon, y + \epsilon]}{range(X) range(Y)}$. The expression for the expected value is similar to order dependency. It will heavily rely on the overlapping of the generated intervals of the domain with the intervals of the real data.

E. Ordered Functional Dependencies (OFD)

An ordered functional dependency combines functional and order dependency and can be defined as follows [18]. It is an OFD, $X \rightarrow Y$, if we consider relation R and attributes $X, Y \subset R$, and $\forall$ tuples $t, u \in R$, if $t[X] = u[X] \rightarrow t[Y] = u[Y]$ but also satisfying $t[X] < u[X] \implies t[Y] < u[Y]$.

Privacy Discussion of Ordered Function Dependency. This relationship imposes more significant restrictions compared to functional and order dependencies. Consequently, the mapping generation adheres to a continuous time Markov chain model [17], facilitating the modeling of dependency transitions. This model is characterized by a transition matrix that specifies state change probabilities; the state changes are similar to the map for ordered functional dependencies. The target states correspond to the domain values of attribute B, rendering the mapping generation akin to a time-variant, one-dimensional directed random walk [23].

The probability of being correct depends on the transition probabilities, and it turns from making a high-dimensional choice to a binary one. But similar to other dependencies, the correctness will still follow a binomial distribution, except the probability for the dependency is time-dependent; thus, the expected value of the number of correct relations generated is given by $N\theta_X\theta_{Y_t}$. A sample probability could be created from a uniform distribution with respect to the total number of remaining partitions $P_{i,i+1} = 1 - \frac{|X| - t}{|Y|}$. This will give us a transition probability of 1 if we have reached the maximum remaining partitions of A so that the relation will be preserved as all values in the domain need to be covered. Following the argument in order and differential dependency, the distribution within the attributes is unknown, meaning that we will also assume the transition probability will be uniform.

The derivations allow us to conclude the data privacy implications of the analyzed dependencies. The focus has been on the implications on data privacy of functional and relaxed functional dependency in conjunction with feature name and domain.

TABLE III
PRIVACY LEAKAGE OF CONTINUOUS ATTRIBUTES

Dep	Attr 0	Attr 2	Attr 4	Attr 5	Attr 6	Attr 7	Attr 8	Attr 9
Rand Gen	580.49	1169.96	0.43	114.17	10.14	138.69	1.71	0.93
Func Dep	580.25	1172.4	0.43	114	10.11	138.6	1.71	NA
Ord Dep	581.43	1383.86	0.24	17.33	9.63	139.44	1	1.41
Num Dep	708.58	NA	NA	NA	NA	NA	NA	NA

Summary

- Metadata, such as the domain of an attribute, enables random generation with the risk of privacy leakage.

- Metadata such as functional and relaxed functional dependencies can be communicated without extra privacy leakage.

TABLE IV
PRIVACY LEAKAGE OF CATEGORICAL ATTRIBUTES

Dependency	Attr 1	Attr 3	Attr 11	Attr 12
Random Generation	44	44	33	44
Functional Dep	44.082	43.954	32.815	NA
Order Dep	44	32	29	47
Numerical Dep	56	NA	NA	NA

V. EVALUATION

This section shows the preliminary experimental evaluation of our main contribution: privacy analysis over relaxed functional dependencies. The evaluation will be bifurcated based on data types: categorical and continuous, as delineated earlier.

Dataset. To assess the practical utility of our solution, we test over a commonly used dataset, echocardiogram [4] from the FD/RFD repeatability project[2]. We chose the echocardiogram dataset as we can discover functional dependencies, order dependencies, and numerical dependencies from this dataset. From other datasets, we can only discover trivial dependencies or oversimplified mappings, which do not serve the purpose of our discussion. Echocardiogram contains 132 rows and 13 attributes. We present the results of the categorical attributes in Table IV and those of the continuous attributes in Table III. We have performed the validation based on the definitions for privacy leakage from Section II. The validation metric will be exact matching and the mean squared error (MSE) for categorical and continuous attributes. All generations derive from the predefined dependencies. The dependencies form a directed graph between the attributes which is used for generation. A fundamental assumption underlying our methodology is that the distribution remains undisclosed. The original raw data will remain in possession of the original possessor.

Table IV shows the number of positive matches for each method using random generation or with functional dependencies, order dependencies, or numerical dependencies. Among the four attributes, values expressed by NA are the attributes that were not discovered for the specific dependency; thus, the dependency cannot be utilized for generating the attribute. The results of positive matches with functional dependencies and order or numerical dependencies are close to a random generation, indicating these dependencies add little value if a malicious party tries to generate a syntactic dataset to mimic the real dataset. Such an observations are consistent with our derivations and conclusions from Section III-B and IV.

The precise index of the appropriate generation may not be critically important, contingent upon the context. To illustrate, consider a collaboration between a financial institution and an e-commerce platform with the goal of loan approval classification. The latter may aim to implement targeted advertising, which does not align with the primary objective. The accurate generation of information can culminate in more effective targeted advertisements compared to no advertisements. Operating under the presumption that all data values are accurate, a proportion of recommendations will facilitate appropriate recommendations. Conversely, inaccurate advertisements yield results comparable to scenarios without available data.

Table III shows the MSE of the generated attribute values against the real values for the discovered dependencies. The MSE is the mean error over many generation rounds to decrease the variance of the error. Following our assumptions, this would be the expected value over this dataset. Different data ranges will, of course, lead to very different MSE values. We can interpret these values as an indicator of a value of $epsilon$ to indicate leakage. Comparing the values in the table, we can conclude that the privacy leakage caused by having dependencies is not more than guessing. Because a larger MSE and fewer exact matches mean less privacy leakage. The outcomes for FDs and RFDs corroborate our prior discussions regarding continuous variables. It is more evident for the continuous variables regarding certain RFDs within datasets.

VI. CONCLUSION

Our examination addressed various metadata characteristics of datasets pertinent to the preparation phase for VFL. The metadata we discussed are feature names, feature domains, functional dependencies, and relaxed functional dependencies. Our findings indicate that specific amalgamations of metadata can precipitate privacy breaches. Notably, privacy leakage was observed to be analogous to instances where feature names, domains, and types were disclosed. We have shown that functional and relaxed functional dependencies do not leak more privacy. We can conclude that feature names and dependencies should be communicated without the domain and type. We have shown that RFDs can be communicated as metadata for VFL without leaking more privacy.

[2]https://hpi.de/naumann/projects/repeatability/data-profiling/fds.html

REFERENCES

[1] General Data Protection Regulation (GDPR) – Official Legal Text — gdpr-info.eu. https://gdpr-info.eu/. [Accessed 01-11-2023].

[2] Information privacy act 2009. https://www.legislation.qld.gov.au/view/pdf/inforce/current/act-2009-014. (Accessed on 08/31/2023).

[3] Personal information protection law of the people's republic of china. http://en.npc.gov.cn.cdurl.cn/2021-12/29/c_694559.htm. (Accessed on 08/31/2023).

[4] Echocardiogram. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C5QW24.

[5] Health Insurance Portability and Accountability Act of 1996 (HIPAA) — CDC, Nov 1996.

[6] Tobias Bleifuß, Susanne Bülow, Johannes Frohnhofen, Julian Risch, Georg Wiese, Sebastian Kruse, Thorsten Papenbrock, and Felix Naumann. Approximate discovery of functional dependencies for large datasets. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1803–1812, New York, NY, USA, 2016. Association for Computing Machinery.

[7] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. Conditional functional dependencies for data cleaning. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 746–755, 2007.

[8] D. Burago, Y. Burago, and S. V. Ivanov. A course in metric geometry. *Graduate Studies in Mathematics*, 2001.

[9] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. Relaxed functional dependencies—a survey of approaches. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):147–165, 2016.

[10] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36(6):87–98, nov 2021.

[11] Aloni Cohen. Attacks on deidentification's defenses. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1469–1486, Boston, MA, August 2022. USENIX Association.

[12] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. Blindfl: Vertical federated machine learning without peeking into your data. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 1316–1330, New York, NY, USA, 2022. Association for Computing Machinery.

[13] Ykä Huhtala, Juha Kärkkäinen, Pasi P. Porkka, and Hannu (TT) Toivonen. Tane: An efficient algorithm for discovering functional and approximate dependencies. *Comput. J.*, 42:100–111, 1999.

[14] Jyrki Kivinen and Heikki Mannila. Approximate inference of functional dependencies from relations. *Theoretical Computer Science*, 149(1):129–149, 1995. ICDT.

[15] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances and challenges. *arXiv preprint arXiv:2211.12814*, 2022.

[16] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature inference attack on model predictions in vertical federated learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 181–192, 2021.

[17] A. A. Markov. The theory of algorithms. *Journal of Symbolic Logic*, 18(4):340–341, 1953.

[18] Wilfred Ng. Ordered functional dependencies in relational databases. *Information Systems*, 24(7), 1999.

[19] Raghu Ramakrishnan, Johannes Gehrke, and Johannes Gehrke. *Database management systems*, volume 3. McGraw-Hill New York, 2003.

[20] J.A. Rice. *Mathematical Statistics and Data Analysis*. Advanced series. Cengage Learning, 2007.

[21] Dan Simovici, Dana Cristofor, and Laurentiu Cristofor. Impurity measures in databases. *Acta Informatica*, 38, 10 2002.

[22] Shaoxu Song and Lei Chen. Differential dependencies: Reasoning and discovery. *ACM Trans. Database Syst.*, 36(3), aug 2011.

[23] H. M. Taylor and S. Karlin. Markov chains. *An Introduction to Stochastic Modeling*, pages 79–163, 2011.

[24] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[25] Daisy Zhe Wang, Xin Luna Dong, Anish Das Sarma, Michael J. Franklin, and Alon Y. Halevy. Functional dependency generation and applications in pay-as-you-go data integration systems. In *International Workshop on the Web and Databases*, 2009.

[26] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM TIST*, 10(2), jan 2019.