Financial Data Exchange with Statistical Confidentiality: A Reasoning-based Approach

Luigi Bellomarini
Banca d’Italia

Rosario Laurendi
Banca d’Italia

Livia Blasi
Banca d’Italia

Emanuel Sallinger
TU Wien and University of Oxford

ABSTRACT
Confidentiality is a crucial requirement in financial data exchange processes. On the one hand, rich microdata is needed for most AI applications, including banking supervision, anti-money laundering, etc. On the other hand, organizations may not be legally authorized to see particular data, e.g., personal data. Striking the right balance provides a number of challenges.

Motivated by our experience with the Central Bank of Italy, in this work we present Vada-SA, a reasoning-based framework for financial data exchange with statistical confidentiality. We present a production-ready and fully engineered framework, adopting a reasoning approach. The framework includes explicit consideration of the reasoning process, the business context and declarative transparency that puts the user in control. We show and discuss a number of risk measures and anonymization criteria, implemented and operated in practice.

1 INTRODUCTION
Confidentiality in financial data exchange has multiple facets and touches different business segments of the FinTech area. In open banking settings, where the increasingly frequent interactions between financial intermediaries motivated by the unbundling and rebundling of the banking process sees the interplay of many actors, each interested in utilizing the data about a specific portion of the process, but with limited or no access-rights to the identity of the involved customers; in European-level banking supervision, where data exchange between the European Central Bank and the National Central Banks needs to reveal situations that are highly critical in terms of the "financial health" of the banks, while the identity of the involved customers tends to be irrelevant; in anti-money laundering, where most modern approaches pinpoint fraudulent or collusive cases by inspecting high-level features of the considered actors, without accessing their identity before any judicial or law-enforcement action authorizes it; in statistical and economic research, with the more and more common establishment of national "Research Data Centers". In these situations, financial authorities require a confidential transfer of personal data a central concern in Europe.

As a matter of fact, financial and statistical authorities and intermediaries look at solutions to share their own microdata, i.e., non-aggregated data at the finest level of granularity, while striking a good balance between their statistical relevance and the need to eliminate any possible trace of personal identities. Many situations arise in the financial segment in which a counterparty must at the same time see parts of the data (to carry out a portion of the process) and must not see other parts which they are not legally authorized to see, e.g., personal data.

This paper is motivated by our experience with the Central Bank of Italy, which, in its capacity of national central bank, banking supervision and oversight authority and Financial Intelligence Unit for Italy, is touched by the problem of confidential financial data exchange in all its perspectives. In this work, we present Vada-SA, the joint effort of the Applied Research Team of the Bank of Italy, TU Wien and the University of Oxford towards a reasoning-based approach to the problem.

The desiderata. We start by laying out the main desiderata for a state-of-the-art financial data exchange solution with confidentiality: (i) It should be context-aware and take into consideration the specific business domain and the characteristics of the involved entities and features to evaluate the risk of a breach of confidentiality; (ii) At the same time it should be schema-independent, and operate regardless of the specific dataset structure; (iii) It should be preemptive, in the sense that it should be able to analyze a given dataset to be exchanged and provide a confidentiality score beforehand, so that analysts can evaluate the risk of sharing it; (iv) It should be active, in the sense that whenever the confidentiality score is over a certain threshold (e.g., statistically inferred or defined by the domain experts), the solution should be able to alter the data and anonymize them so that the threshold is respected; (v) It should embody a statistics-preserving anonymization logic, by removing the minimum amount of information needed to guarantee confidentiality, while preserving the statistical soundness and relevance of data; (vi) It should be fully explainable, meaning that the confidentiality score of a candidate dataset as well as the reasons for specific anonymization choices should be completely understandable to domain experts; plus it should have a transparent semantics of confidentiality; (vii) It should be business-friendly, by being extensible, IT-independent and at business level, i.e., domain experts should operate autonomously in defining new scoring criteria as well as anonymization logic in a high-level non-technical language; (viii) It should be scalable and able to handle increasing data volumes.

Statistical Disclosure Control. The area of Statistical Disclosure Control [26, 35, 37] (SDC) represents a relevant yardstick for our work. The SDC approach concentrates on re-identification, i.e., the possibility for an attacker to cross-link information it rightfully retains, in particular, every single tuple of a legitimately owned database, with other data sources so as to find out the underlying identities (of the involved people, companies and stakeholders in general). SDC adopts quantitative indicators to take decisions on data sharing by evaluating the risk of re-identification and balancing it with the measure of the statistical...
relevance of the data, so as to minimize the risk while maximizing the statistical utility. SDC also studies solutions to transform, namely anonymize, the data to be shared, balancing confidentiality and statistical relevance. Commonly adopted techniques, featured by widespread tools such as sdcMicro [9], μ-ARGUS [27], and ARX [33], aim at removing potential identifiers (sometimes known as quasi-identifiers) of the disclosed tuples and include value suppression, aggregation, and generalization.

Unfortunately, the approaches and the tools offered so far by the SDC community do not fulfill the desiderata of a full-fledged solution needed by the processes of financial companies and organizations, like the Bank of Italy. First, to the best of our knowledge, all the existing SDC techniques are schema dependent and anonymization risk assessment and anonymization programs are tightly coupled to the dataset structure. Then, SDC techniques are only based on value statistics within the dataset to be anonymized and are not context-aware, while it is our experience that the risk of disclosure highly depends on the characteristics of the source and target databases [18] as well as the surrounding business information, e.g., availability of specific cross-linking data, even at tuple level. As a consequence, SDC techniques tend to fall short of accuracy in this respect. Although the anonymization techniques of SDC put into action interesting ideas and, in general, preserve statistical relevance of the datasets, to the best of our knowledge, all of them lack full explainability, unacceptable for financial organizations with strong accountability constraints. The lack of explainability prevents effective feedback-based adaptivity and the improvement of disclosure control proceeds by trial and error. Furthermore, all the existing tools tend to be not business friendly: they adopt a technology- and IT-dependent language (e.g., R libraries or Java), often lack clear semantics (typically only informally explained in the documentation), require adopters to have a technical background and are hardly extensible. Finally, such tools are data-scientist oriented libraries and, while showing good performance, do not have formal scalability guarantees.

**Contribution.** In this work we present VADA-SA, a reasoning-based framework for financial data exchange with statistical confidentiality. It is based on our long-term experience in developing AI-enhanced data-driven solutions revolving around logic-based reasoning. In particular, this work builds on the Vadalog System [6], a state-of-the-art reasoning system leveraging the Vadalog language, a member of the Datalog\(^6\) family [12], exhibiting very good characteristics of scalability and expressive power. In particular, we contribute as follows.

- **We present a production-ready and fully engineered framework, VADA-SA, for financial data exchange with confidential privacy, adopting a reasoning approach.** The enterprise data to be shared, along with the metadata, are modeled as the extensional component of the reasoning process, whereas standard risk measures and anonymization methods are modeled as the intensional component of the process, i.e., a set of Vadalog rules. The activation of the rules upon the extensional component—i.e., the reasoning process—produces the derived extensional component, which is either a fully explained risk measure for a given dataset to be exchanged or its anonymized version.

- **We show and discuss (and through VADA-SA ship off-the-shelf) a number of risk measures and anonymization criteria and illustrate how they can be managed in Vadalog.**

- **We suggest that the surrounding business context relevant for accurate risk measures is awarly modeled within the intensional component in terms of Vadalog rules, which are at the same time schema independent w.r.t. the structure of the datasets.** Although the framework targets financial data as a primary application, the techniques we present are general and can be applied in any context requiring statistical confidentiality.

- **We envision that the SDC techniques can be used as a solid theoretical basis to craft a statistically preserving anonymization logic, yet, unlike existing approaches, we model in a purely declarative way in terms of Vadalog rules.**

- **We embrace a user-delegation approach, in the sense that by means of a semantically clear, fully declarative, non-technical and IT-independent language (i.e., characteristics that Vadalog embodies by design [6]), we delegate specific users to writing their own criteria and encoding the business knowledge, with cost and operational savings.**

- **In our framework, we inherit a set of benefits from logic-based reasoning. In particular, we refer to the pros of declarative approaches that, unlike procedural programming, relieve the users from the need to understand the internals of anonymization methods when adopting it. Full explainability is guaranteed by standard logic entailment semantics, enforced with chase-based procedures [20] embodied in VADA-SA. Finally, the ideal balance between computational complexity and expressive power inherited from Vadalog, allows VADA-SA to achieve very good scalability.**

- **We discuss an interesting set of real-world risk measures and anonymization criteria, implemented and operated in practice.**

**Overview.** The remainder of the paper is organized as follows. In Section 2 we pursue the industrial setting at the Bank of Italy. In Section 3 we introduce the background about Vadalog. Section 4 presents the VADA-SA framework and Section 5 shows it in action in relevant cases from the Bank of Italy. In Section 6 we discuss some related work and Section 7 concludes the paper.

## 2 INDUSTRIAL SETTING

The Bank of Italy has recently set up a Research Data Center (RDC).\(^1\) At its core, there are a set of relational databases that store the microdata, i.e., the operational finest-grained data, from many core business applications such as the credit risk register, payment systems, balance of payments, banking supervision indicators, etc. The ultimate goal of RDC is sharing statistically relevant information with other cooperating institutions such as the National Statistical Office, other central banks, the European Central Bank, universities and research centers. While all these counter-parties operate within a “circle of trust”, and can thus access the mentioned microdata, the identities of the involved entities, be they companies, banks or people, should remain of the sole responsibility (and therefore visibility) of the Bank of Italy, which is legally in charge of the respective processing.

The microdata that the RDC deals with regard different business processes and originate from multiple sources, usually external to the Bank of Italy. These data are collected with a variety of methods such as statistical surveys or data flows and are organized into several microdata DBs, by business domain. The RDC aims at including 65 microdata DBs with operational data from 1977 to 2020 and expected size of 30-50TB, with a 1TB/month growth. The RDC currently stores 14 microdata DBs, about families and individuals, firms, and historical data, including:

- Household income and wealth
- Household finance and consumption

\(^{1}\)The RDC is part of the INEXDA initiative (http://www.inexda.org/) for the exchange of granular statistical data.
### Figure 1: Microdata DB about inflation and growth.

- Financial literacy data
- Business outlook of industrial and service firms
- Italian housing market
- Inflation growth and expectations
- Historical archive of Italian credit.

Microdata DBs contain business data, including attributes that may disclose, directly or indirectly, the identity of the involved subjects; let us call these subjects respondents, by some abuse of the terminology adopted for statistical surveys. The risk for a tuple of a microdata DB to be associated (i.e., ‘linked’) to the respective real-world identity of the respondent is named risk of re-identification. Indeed, the notion of re-identification revolves around the (realistic) assumption that an external data source containing all the identities of the respondents exists; let us call identity oracle such database. The challenge here consists in mitigating the risk that an attacker could be able to link the value of some attributes of a tuple of the microdata DB, with those of a single tuple (or a very small set thereof) of the identity oracle and therefore disclose the respondent’s identity.

### 2.1 Setting Foundations

Let us frame our industrial context with the needed foundations.

**Relational Foundations.** Let $C$, $N$, and $V$ be disjoint countably infinite sets of constants, (labelled) nulls and (regular) variables, respectively. $A$ (relational) schema $S$ is a finite set of relation symbols (or predicates) with associated arity. A term is either a constant or variable. An atom over $S$ is an expression of the form $R(\bar{v})$, where $R \in S$ is of arity $n \geq 0$ and $\bar{v}$ is an $n$-tuple of terms. A database instance (or simply database) over $S$ associates to each relation symbol in $S$ a relation of the respective arity over the domain of constants and nulls. The members of relations are called tuples. By some abuse of notations, we sometimes use the terms tuple and fact interchangeably.

**The Microdata DB and the Identity Oracle.** A microdata DB is a relation of schema $M(I, \bar{q}, \bar{z}, W)$, where $I$ is an $n$-tuple of attributes defined as direct identifiers, $\bar{q}$ is an $n$-uple of quasi-identifiers, $\bar{z}$ is a set of non-identifying attributes and $W$ is a sampling weight. An identity oracle is a relation of schema $O(\bar{t}, \bar{q}'$, $I)$, where $\bar{t}$ is a set of direct identifiers, $\bar{q}'$ is a set of quasi-identifiers and $I$ is the identity of the respondent.

- **Direct identifiers** are attributes s.t. their values (of each single attribute, separately) allow to determine the identity of the respondent, that is, for a given tuple of $M$, the join between $M$ and $O$ on an attribute of $I$ equated to an attribute of $\bar{t}$ selects a single tuple from $O$ and therefore the resulting tuple discloses the respondent’s identity. Observe that a direct identifier is a key attribute for $O$ and it is assumed that $I \subseteq \bar{t}$. Examples of direct identifiers are the social security number, the Italian fiscal code, the driving licence number, etc.

- **Quasi-identifiers** are attributes s.t. the values of two or more of them, jointly, are likely to disclose the identity of the respondent, that is, for a given tuple of $M$, the join between $M$ and $O$ on two or more attributes of $\bar{q}$ equated to attributes of $\bar{q}'$ selects a small set of tuples of $O$ and therefore likely discloses the respondent’s identity. In other terms, quasi-identifiers are features that in specific combinations are enough selective to endanger the respondent’s confidentiality. This selectivity depends on the attribute (as some are intrinsically more specific) and, of course, on the combination of values, which can be more or less specific for a given context. For example, the joint use of age and address can be quite selective if we refer to a context of small dwellings, whereas gender and address would be less selective. On the other hand, occupation-gender is in general not very selective, whereas it can be extremely discriminating if we are referring to a context of a survey about gendered jobs in some country.

- **Non-identifying attributes** are those that do not fall in the two previous categories. These attributes are not critical because neither individually nor in combination with others, allow to disclose the identity of the respondent, i.e., re-identification is not possible. On the one hand, this can depend on an intrinsic scarce selectivity of the attribute like in the case of age in a given context, on the other hand, a non-identifying attribute can be even intrinsically identifying, yet its value certainly unknown to the identity oracle. This is the case, for instance, of internal system identifiers which are useless for re-identification.
• **Context and sampling weight.** We have touched on the notion of context when discussing quasi-identifiers, which are more or less selective depending on the domain of discourse. The context can be seen as a selection of tuples from $O$ based on the domain of interest. For instance, if we were surveying the population of Milan, the only tuples of $O$ referring to people living in Milan could be used to attempt re-identification of tuples of $M$, thus making it easier. The sampling weight accounts for the context by measuring the representativeness of a tuple $t$ of $M$ w.r.t. the entire context $C$ to which $M$ refers. In this sense, $R$ is a sample from $O$, and $W_t$ is the tuple sampling weight.

There are different options for defining the sampling weight [7, 22]. The one we take inspiration from is the expected value of the number of entities having the same characteristics as $t$ (according to a similarity function $\phi$) in the sample distribution of $O$ according to a given context $C$. Given $M$, the weight $W_t$ can be estimated for each tuple $t$ from the posterior distribution of values for $\hat{q}$ among the tuples. Many options are also possible for $\phi$ and the simplest one just uses equality of quasi-identifiers attributes. Higher weights denote statistically relevant tuples, likely carrying scarcely selective attributes; lower weights denote statistically less relevant tuples (outliers, as a limit case), likely with highly selective attributes.

• **Identity.** The value of such an attribute stands for some universally recognized representation of one respondent’s identity.

In our experience with the Bank of Italy, the categorization of microdata DB attributes as direct, quasi- and non-identifiers as well as weight estimation is a hybrid process involving human experience-based evaluation, learning from training sets, and domain-based reasoning, as we shall see.

### 2.2 Towards a Reasoning Framework for Statistical Disclosure Control

With the depicted context, we can achieve a straightforward definition of re-identification risk as the probability $p_t = 1/W_t$ of re-identifying $t$ given the value of all its quasi-identifiers $\hat{q}$. We can say that, in some sense, provided that $O$ is an abstraction, the sampling weight $W_t$ is an estimator for the cardinality of the join $|\sigma_t(M) \bowtie O|$, where $\bowtie$ denotes the selection and $\bowtie$ the join.

However, re-identification risk is an upper bound for the real disclosure risk, in the assumption that all quasi-identifiers are known to a potential attacker. As a matter of fact, for a given tuple we may be interested in evaluating the risk only w.r.t a subset $\hat{q} \subseteq \hat{q}$ of quasi-identifiers, the ones we suppose the attacker is aware of or are more selective. Moreover, we may want to apply an arbitrary risk weight function $\lambda$, which takes as input $W_t$ as well as the values for quasi-identifiers of $t$. Whence, the following definition of a general statistical disclosure risk:

$$\rho_q = 1/\lambda(\sigma_q(M))$$

The function $\lambda$ computes an aggregate weight over the tuples selected by $\hat{q}$ and generalizes many different risk measurement techniques, as we shall see, including the re-identification-based risk (for which $\lambda(\sigma_q(M)) = \sum_{t \in \sigma_q(M)} (W_t)$), but also $k$-anonymity [34], individual risk [7], and SUDA [19].

**Use cases.** Our industrial goal consists in: 1. evaluating the statistical disclosure risk and, 2. if unacceptable, take actions to counteract possible information disclosure, while preserving statistical significance (anonymization). The joint performance of the two mentioned actions is known as statistical disclosure control [18].

![Figure 2: The attack strategy in action: by querying the identity oracle along Area, Sector and number of Employees, an attacker can narrow the search to few candidates and make a plausible guess about respondents’ identity.](image-url)
operate on less representative tuples so as to increase overall confidentiality without hampering the statistical significance.

3 VADALOG REASONING

Vada-Sa, the statistical disclosure control framework we introduce in this paper, is based on the Vadalog system, a state-of-the-art logic-based reasoner [6] whose core revolves around the Vadalog language, a member of the Datalog family [12, 23]. The disclosure risk measurement techniques as well as the anonymization logic are expressed in Vadalog.

Datalog\(^\pm\) generalizes Datalog with existential quantification in the rule conclusion, making it suitable for ontological reasoning. A rule is a first-order sentence of the form \(∀\bar{x}∃\bar{y}[\varphi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}[\psi(\bar{x}, \bar{z})]\) where \(\varphi\) (the body) and \(\psi\) (the head) are conjunctions of atoms. For brevity, we omit universal quantifiers and denote conjunction by comma. As usual in this context, the semantics of a set of rules is operationally defined by the well-known CHASE procedure. Intuitively, the CHASE satisfies the rules by generating new head facts for bindings of the body, possibly introducing new variable symbols in the data, in the form of labelled nulls, in the presence of existentially quantified head variables [21].

The core of Vadalog is based on Warded Datalog\(^\pm\) [6], a syntactic restriction to Datalog\(^\pm\) that guarantees decidability and tractability in the presence of recursion and existential quantification. In terms of expressive power, Warded Datalog\(^\pm\) captures full Datalog and OWL 2 direct semantics entailment regime for OWL 2 QL. The language underpinnings are exploited by the reasoner to allow for efficient execution of reasoning tasks. Vadalog augments Warded Datalog\(^\pm\) with supplementary features such as aggregation, algebraic operations, and stratified negation. As we shall see, Vadalog is sufficiently expressive to support our anonymization reasoning scenarios and comprises all the needed features such as joint use of full recursion, existential quantification and aggregation, to model propagation of the disclosure risk and anonymize values. These requirements are not met by the standard relational/SQL systems, which in particular offer inefficient or no support for recursion and existentials.

4 THE VADA-SA FRAMEWORK

While statistical disclosure control has traditionally followed a procedural approach, we propose a shift towards a fully declarative one, and look at state-of-the-art reasoners, leveraging our experience on Vadalog and Knowledge Graphs applied to different problems in the financial realm e.g., link prediction [2] as well as schema-independent approaches to model management [3]. The VADA-SA framework, whose architecture is sketched in Figure 3, lies on the following basic pillars:

- A structuring of the statistical disclosure control process in the form of an anonymization cycle.
- The construction of a Vadalog-based enterprise Knowledge Base (KB) encompassing the patterns and techniques for statistical disclosure risk assessment and anonymization as well as all the surrounding business knowledge to be leveraged.
- The formulation of risk assessment and anonymization phases in the form of reasoning tasks upon the KB. In such reasoning tasks, the extensional component comprises the microdata DB as well as their basic metadata, such as schema-level information. Much care is devoted to the intensional component, encoding reasoning rules for: attribute categorization, risk assessment and anonymization. The intensional component is at high level of abstraction, composed of pluggable Vadalog modules, some of which are provided off-the-shelf while others can be autonomously developed by business experts. The overall statistical disclosure control process is a reasoning task itself, which relies on the mentioned ones and adaptively chooses the actions to be performed. The derived extensional component, i.e., the results of the reasoning process, contains the outcome of risk analysis and the anonymized microdata DBs.

In this section, we first illustrate the anonymization cycle and the metadata dictionary, at the basis of our schema-independent approach (Section 4.1), we then focus on the evaluation of statistical disclosure risk (Section 4.2) and anonymization (Section 4.3). Finally, we show extensions and advanced applications involving complex business knowledge (Section 4.4).

4.1 The Anonymization Cycle and the Metadata Dictionary

When a microdata DB needs to be shared, it undergoes the anonymization cycle at the core of the VADA-SA architecture, shown in Figure 3. It consists of an iterative application of disclosure risk evaluation and anonymization until the risk is under a given threshold. Each iteration removes a minimum amount of information, and checks whether confidentiality requirements are fulfilled, in a statistics-preserving fashion.

In particular, risk evaluation takes as input a microdata DB. Based on its category, each attribute has a different treatment. Direct identifiers must not be disclosed and non-identifying attributes are not needed in the risk evaluation process, thus both are dropped. Quasi-identifiers and the sampling weight are used for disclosure risk estimation. Anonymization is activated until the disclosure risk is acceptable. In so doing, we aim at a trade-off between statistical preservation and disclosure risk, as captured by the threshold \(T\), determined on the basis of user experience.

Metadata Dictionary and Attribute Categorization. In order to achieve schema and data independence, in VADA-SA we follow a meta-level approach and include a metadata dictionary in the KB. Facts of the form MicroDB(name, Attr(MicroDB, name, description), Category(MicroDB, attr, cat)) are used to reason upon microdata DBs, their attributes and their categories, respectively. Figure 4 shows the portion of Vada-Sa dictionary for the "I&G" (Inflation and Growth) microdata DB. Facts for MicroDB and Attribute are part of the extensional component and change when new microdata DBs are added into VADA-SA. Facts for Category are part of the derived extensional component: they are the product of a reasoning process that, for each microdata DB and each attribute, infers the most suitable category. In fact, before entering the anonymization cycle, the attributes of the microdata DB need to be categorized as identifiers, quasi-identifiers or non-identifying attributes, as we have seen in Section 2.1.
Algorithm 1 Attribute categorization

(1) $\text{Att}(M, A) \rightarrow \exists C \text{ Cat}(M, A, C)$. $\text{Cat}(M, A, C)$. $\text{ExpBase}(A, C)$. $\text{Cat}(M, A, C) \rightarrow \text{ExpBase}(A, C)$. $\text{Cat}(M, A, C_1), \text{Cat}(M, A, C_2) \rightarrow C_1 = C_2$.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I&amp;G Id</td>
<td>Company Identifier</td>
</tr>
<tr>
<td>I&amp;G Area</td>
<td>Geographic Area</td>
</tr>
<tr>
<td>I&amp;G Sector</td>
<td>Product Sector</td>
</tr>
<tr>
<td>I&amp;G Employees</td>
<td>Num. of employees</td>
</tr>
<tr>
<td>I&amp;G Residential Rev.</td>
<td>Rev. from internal market</td>
</tr>
<tr>
<td>I&amp;G Export Rev.</td>
<td>Rev. from external market</td>
</tr>
<tr>
<td>I&amp;G Export to DE</td>
<td>Rev. from DE market</td>
</tr>
<tr>
<td>I&amp;G Growth</td>
<td>Growth last 6 mths</td>
</tr>
<tr>
<td>I&amp;G Weight</td>
<td>Sampling Weight</td>
</tr>
</tbody>
</table>

Algorithm 2 Anonymization cycle

(1) $\text{Val}(M, I, A, V), \text{Cat}(M, A, C), C \in \{\text{Quasi-identifier}, \text{Weight}\}, V\text{Set} = \text{munion}(A, V) \rightarrow \text{Tuple}(M, I, V\text{Set})$. $\text{Tuple}(M, I, V\text{Set})$. $\text{Tuple}(M, I, V\text{Set})$, $\text{risk}(I, R), R > T \rightarrow \text{#anonymize}(I)$. $\text{Tuple}(M, I, V\text{Set})$, $\text{risk}(I, R), R \leq T \rightarrow \text{Tuple}_A(M, I, V\text{Set})$.

The set of VADALOG rules of Algorithm 2 compactly represents this logic. Rule (1) creates Tuple facts for each tuple of the microdata DB $M$, identified by an artificial identifier $I$, and collects all the name-value pairs for quasi-identifiers and sample weights into the VSet variable. Val facts are part of the extensional component and store the value $V$ for an attribute $A$ of the microdata DB $M$. The identifiers of $M$ are implicitly dropped. Observe that $\text{munion}$ performs such aggregation, for each microdata DB $M$ and VSet is a set-type variable. Whenever a specific tuple $I$ violates a $[0, 1]$ risk threshold $T$, a fact anonymize is produced for $I$, in Rule 2. Both $\text{risk}$ and $\text{anonymize}$ are atoms defined in external libraries, in VADALOG (denoted by the "#" prefix). In particular, $\text{risk}$ returns the risk $R$ associated to a given tuple $I$; it is a compact form for the join "$\text{riskInput}(I), \text{riskOutput}(I, R)$", where $\text{riskInput}$ is a fact triggering a VADALOG program producing facts of $\text{riskOutput}$ for $I$. More simply, $\text{anonymize}$ produces new facts for Tuple. This mechanism embeds a recursion on Rule 2, to anonymize tuples that still do not pass the risk evaluation. Only those facts for Tuple that pass the risk validation of Rule 3 are copied to Tuple$_A$, which can be considered anonymized.

The anonymization cycle in Algorithm 2 makes the approach fully explainable in the sense that each anonymization decision taken by Rule 2 is motivated by the specific binding of its body. It is also preemptive and active, in the sense that for each threshold violation, greedily applies a single anonymization step, at the same time minimizing the amount of suppressed statistical information. It is schema independent, as only atoms of the metadata dictionary are used and there is no specific reference to either schema or instance objects of the single microdata DBs. We shall see how specific values are bound to the attributes as a responsibility of $\text{risk}$ and $\text{anonymize}$ implementations. Finally, the algorithm offers multiple degrees of freedom: different risk and anonymization techniques can be used, and our $\text{risk}$ and $\text{anonymize}$ and polymorphic, in this sense: specific optimizations and execution heuristics can be adopted to choose which tuples to anonymize first (by controlling the activation order of Rule 2 against its possible bindings), and to choose which quasi-identifiers to anonymize first.

4.2 Statistical Disclosure Risk Estimation

Our $\text{risk}$ atom in Algorithm 2 is polymorphic. VADA-SA features a plug-in mechanism to opt for specific implementations at runtime. While the high-level characteristics of the VADALOG language allow to delegate users to specify their own risk logic, with extensive use of business knowledge, a number of techniques are provided off-the-shelf. In this section, we introduce the main risk disclosure evaluation techniques offered by VADA-SA.

Re-identification-based. We start in Algorithm 3 with re-identification-based risk evaluation, that we have defined in Section 2.2.

Algorithm 3 Re-identification-based risk evaluation

(1) $\text{Tuple}(M, I, V\text{Set}), \text{riskInput}(I), \text{Cat}(M, W, \text{Weight}), R = 1/\text{msum}(V\text{Set}[W], (I)) \rightarrow \text{Tuple}_A(R, V\text{Set}[\text{AnonSet}])$. $\text{Tuple}(M, I, V\text{Set}), \text{Tuple}_A(R, V\text{Set}) \rightarrow \text{riskOutput}(I, R)$.

Whenever a tuple $I$ needs to be evaluated (riskInput atom), in Rule 1, the name $W$ of the weight atom is retrieved from the metadata dictionary and used to extract the weight value from VSet, with the access operator denoted by $[X]$, where $X$ can be either a single attribute name or a collection thereof. Weights are summed (msum) and the risk score $1/R$ is computed. Tuple$_A$ has
variable arity, and its terms are the computed risk \( R \) and the set of quasi-identifiers to group by when forming the summation. The expression \( \text{VSet[AnonSet]} \) has the following meaning: the prefix operator "\{" is collection unpacking and turns each element of the argument collection into a term of TupleA, essential for grouping along quasi-identifiers. Note that VSet is filtered by a set AnonSet of attribute names — the order is irrelevant in this context— that selects those that are considered of interest by business experts w.r.t. risk evaluation. Rule 2 finds those tuples \( I \) for which the risk has been computed and returns it. It uses the packing operator, denoted by the suffix operator "," which packs a sequence of terms of TupleA into the set variable VSet. In this way, by joining on VSet, we can identify all the tuples to which the risk computation applies. We have already seen examples of re-identification-based risk estimation in Section 2.2.

**k-anonymity** is a commonly used threshold approximation of re-identification risk estimation [34]. For a given set of quasi-identifiers, whenever the number of occurrences is less than a fixed threshold \( k \), it is considered dangerous; it is safe otherwise. For instance, in the microdata DB of Figure 1, considering the quasi-identifiers Area and Sector, we notice, e.g., that there is only one occurrence for "North" and "Public Service" (tuple 1). We say the set of pairs \((\{\text{Area},\text{North}\}, \{\text{Sector,Public Service}\})\) is a sample unique for tuple 1. The \texttt{Vadalog} reasoning rules for k-anonymity are reported in Algorithm 4.

**Algorithm 4 k-anonymity**

1. Tuple(M, I, VSet), riskInput(I), \( R = \text{mcount}(I) \) \( \rightarrow \) TupleA(\( R \), \text{VSet[AnonSet]}).
2. Tuple(M, I, VSet), TupleA(\( R \), VSet) \( R = \text{case} R_1 < k \text{ then } 1 \text{ else } 0 \rightarrow \text{riskOutput}(I, R) \).

**Individual Risk.** In the re-identification model the simplifying assumption is made that the sampling weight \( W_I \) corresponds to the frequency (number of occurrences) \( F_k \) of a given combination \( k \) of quasi-identifiers in the total population from which the microdata DB has been sampled; therefore we can compute the combination risk as \( 1/F_k \). Yet, frequencies \( F_k \) are unknown and in general different from \( W_I \). A further inferential step is then required. The typical approach [7, 22, 38] is accounting for \( F_k \) in a Bayesian fashion, by considering the distribution of the population frequencies given the sample frequencies \( F_k \) and obtaining \( 1/F_k \) as the posterior mean. In our setting, the sample frequency is the sample count in the microdata DB. Different assumptions can be made on the posterior distribution of \( F_k \) with different techniques to accordingly estimate \( \hat{p}_k \). The one we adopt here is considering such distribution a negative binomial and thus we pose \( \lambda = \Sigma W_i/f_i \) to estimate risk in Equation 1 of Section 2.2. Indeed, other distributions can be adopted. The individual risk estimation is formalized in Algorithm 5.

**Algorithm 5 Individual risk**

1. Tuple(M, I, VSet), riskInput(I), Cat(M, W, Weight), \( F = \text{mcount}(I) \), \( R = \text{msum}(\text{VSet}[W], I) \) \( \rightarrow \) TupleA(F/R, \text{VSet[AnonSet]}).
2. Tuple(M, I, VSet), TupleA(R, VSet) \( \rightarrow \text{riskOutput}(I, R) \).

While scanning through the tuples having a given combination VSet, used as a group-by key thanks to the unpacking operator, Rule 1 counts the occurrences of each combination (frequency) and sums the contributor weights. Facts for TupleA are produced only once all the contributors are available. The risk is then estimated for each combination and finally returned by Rule 2.

**SUDA.** With k-anonymity, we have introduced the concept of sample unique, i.e., a set of quasi-identifiers — name-value pair— that identify a tuple of a given microdata DB, i.e., they are unique. A sample unique is not the same as a database key, because it expresses a property that holds at tuple level and not at schema level. Alongside the schema-level distinction between superkey and key (a minimal superkey) in relational theory [1], here, at data level, we introduce the minimal sample unique (MSU) \( \mu_k \) for a given tuple, that is a sample unique for which there exists no other sample unique \( \mu_i \) for the same tuple, s.t. \( \mu_i \subset \mu_k \). The Special Unique Detection Algorithm (SUDA) is a heuristic technique that estimates the statistical disclosure risk of a given tuple based on the size and the number of its MSUs.

Consider for example the set \( \mu_{20} = \{\langle \text{Area, Center}\rangle, \langle \text{Sector, Financial}\rangle, \langle \text{Employees, 1000+}\rangle, \langle \text{Res. Rev., 30-60}\rangle\} \) for the microdata DB in Figure 1 for tuple 20. It is sample unique though not MSU, since the set \( \mu_{1000+} = \{\langle \text{Sector,Financial}\rangle\} \) is sample unique and s.t. \( \mu_{20} \subset \mu_{1000+} \). Moreover, \( \mu_{20} \) is MSU. Similarly, \( \mu_{1000+} = \{\langle \text{Employees, 1000+}\rangle, \langle \text{Res. Rev., 30-60}\rangle\} \) is another MSU. In total, tuple 20 has 2 MSUs.

Algorithm 6 encodes the VADA-SA version of SUDA.

**Algorithm 6 SUDA**

1. Tuple(M, I, VSet), riskInput(I) \( \rightarrow \) TupleA(M, I, VSet).
2. Tuple(M, I, VSet), Cat(M, A, Quasi-identifier), \( A \in \text{VSet} \rightarrow \exists Z \text{ Comb}(Z, I), \text{In}(A, Z) \).
3. Comb(Z, I), Tuple(M, I, VSet), Cat(M, A, Quasi-identifier), \( A \in \text{VSet}, \text{not In}(A, Z_1) \rightarrow \exists Z \text{ Comb}(Z, I), \text{In}(A, Z_1) \).
4. Comb(Z, I), Tuple(M, I, VSet), \( A \in \text{VSet} \rightarrow \exists Z \text{ Comb}(Z, I), \text{In}(A, Z) \).
5. Comb(Z, I), Tuple(M, I, VSet), \( A \in \text{VSet} \rightarrow \exists Z \text{ Comb}(Z, I), \text{In}(A, Z) \).
6. TupleC(I, VSet*), \text{mcount}(I) \rightarrow 1 \rightarrow \exists S \text{ Su}(S, VSet), \text{HasSu}(S, I).
7. Su(S, VSet), HasSu(I, S), not HasSu(I, S), \text{Su}(S, VSet*), VSet* \subset VSet \rightarrow \text{MSU}(I, S).
8. Tuple(M, I, VSet), MSU(I, S), Su(S, VSet), \( R = \text{case size}(VSet) < k \text{ then } 1 \text{ else } 0 \rightarrow \text{riskOutput}(I, R) \).

After restricting the focus on input tuples (Rule 1), for each tuple we generate all the combinations of quasi-identifiers, first by introducing a combination Z for each of them (Rule 2), and then by constructing all the possible extensions that can be obtained by adding other quasi-identifiers (Rules 3 and 4). Then, for each combination of quasi-identifiers, generated by unpacking (Rule 5), we generate sample unique facts for Su, denoting those combinations that exactly identify one single tuple. The predicate HasSu is needed since every tuple I can have multiple sample unique sets, while the mcount aggregation needs to group by VSet. Rule 7 creates facts for MSU, filtering only those sample unique sets that are minimal. Finally, Rule 8 implements the logic to handle minimal sample unique sets. In this case, we evaluate the size of every MSU and if it is above a given threshold \( k \), we consider the input tuple dangerous and thus return 1. The assumption here is that we cannot accept that the number of quasi-identifiers that can disclose the identity is too small. Clearly, more sophisticated checks could be implemented, possibly also including an overall evaluation of all the MSUs for a given tuple, for example by comparing the average size of MSUs against a threshold.
containing one or more nulls may match with different tuples of the microdata DB, or even with none of them, depending on the specific assignment for those nulls.

Going back to what introduced in Section 2.2, in our framework in order to estimate the statistical disclosure risk, we need to compute \( \lambda(\sigma_{q_1 \ldots q_k} M) \) over a selection of the microdata DB \( M \), based on an \( n \)-uple of values \( \mathbf{q} \) for quasi-identifiers. The risk estimation techniques of Section 4.2 apply \( \lambda \) to the entire microdata DB and the selection is implicit in the grouping performed by the aggregations, in the sense that, for each tuple, the aggregation forms the group by selecting only those tuples having the same values for quasi-identifiers (or subset of interest, thereof).

So, \( M(1, \mathbf{q}_1, \ldots, \mathbf{q}_n) \) is included in the selection induced by tuple \( M(1, \mathbf{q}_1, \ldots, \mathbf{q}_n) \) iff \( (q_1', \ldots, q_n') = (q_1, \ldots, q_n) \), and, by construction, the groups form a partition of the microdata DB. If we allow \( q_i \) to be a labelled null, a new semantics must be adopted to define whether \( q_i = q_i' \) and thus form the aggregation groups.

The introduction of nulls raises non-trivial semantic issues when aggregations are involved, and theoretical work is still needed to achieve sound characterizations [25]. In VADA-SA, for the construction of aggregation groups, we adopt a null-tolerant semantics inspired by the so-called maybe-match approach [14], and assume that \( q_i =_1 q_i' \) holds if: (i) \( q_i \) and \( q_i' \) have the same constant value, or (ii) either \( q_i \) or \( q_i' \) is a labelled null. Consequently, \( (q_1', \ldots, q_n') = (q_1, \ldots, q_n) \) holds iff \( q_i =_1 q_i' \) holds for every \( 1 \leq i \leq n \). The \( =_1 \) relation is therefore used, instead of standard equality, to form groups. A tuple containing null quasi-identifiers, as a result of anonymization steps, is assigned to multiple aggregation groups (which do not partition the microdata DB anymore), increasing their cardinality and so anonymity.

We now have all the ingredients ready to encode local suppression, an anonymization method where quasi-identifiers are replaced by labelled nulls to reduce the statistical disclosure risk. The technique is expressed by Algorithm 7.

**Algorithm 7 Local suppression**

1. \( 
\begin{align*}
\text{Tuple}(M, I, VSet), \text{ anonymize}(I), \text{Cat}(M, A, \text{Quasi-identifier}), \\
VSet[A] \in \text{not null} \rightarrow 3 \exists Z \text{ Tuple}(M, I, (A, Z) \cup (VSet \setminus (A, _{\ldots})))
\end{align*}
\)

**For a tuple \( I \) that needs to be anonymized, as witnessed by the predicate anonymized, for a not null quasi-identifier \( A \), we generate a new tuple, where it is replaced by a labelled null \( Z \).**

**Global Recoding** While local suppression introduces nulls, another technique to control statistical disclosure risk consists in decreasing the granularity of the values on the basis of domain knowledge. Consider again Figure 5a. Tuples 6 and 7 have the following sample unique sets, respectively: \{ \{Area, Milano\} , \{Sector, Construction\} \}, \{ \{Area, Torino\} , \{Sector, Construction\} \} and therefore have high disclosure risk. Besides the basic metadata dictionary we have seen in Section 4.1, the VADA-SA KB contains knowledge about the attribute domains as well as the mutual relationship between their values. For instance, for the attribute \( \text{Area} \), the KB comprises the following information:

\[
\begin{align*}
\text{Att} & (\text{ISG, Area}), \text{TypeOf} (\text{Area, City}), \text{SubTypeOf} (\text{City, Region}), \\
\text{InstOf} (\text{Milano, City}), \text{InstOf} (\text{Torino, City}), \\
\text{InstOf} (\text{North, Region}), \text{IsA (Milano, North)}, \text{IsA (Torino, North)}.
\end{align*}
\]

The \( \text{Area} \) attribute is known to be of Type “City”, which in turn is a SubtypeOf “Region”. Moreover we know that Milano and Torino are instances of cities and North is an instance of region. Finally, we now that both Milano and Torino are in the North. Similar knowledge is present for the entire geography.
Algorithm 8 Global recoding
(1) Tuple(𝑀, 𝐼, 𝑉Set), anonymize(𝐼), Cat(𝑀, 𝐴, Quasi-identifier), TypeOf(𝐴, 𝑋), subTypeOf(𝑋, 𝑌), \( \rightarrow \) Tuple(𝑀, 𝐼, (𝐴, 𝑍)) \( \cup \) (𝑉Set \( \setminus \) (𝐴, )),

The logic for global recoding is in Algorithm 8. For a tuple that needs to be anonymized, we consider a quasi-identifier \( 𝐴 \). Based on its type, we climb the hierarchy up to its direct super-type \( 𝑌 \). Then for the value \( 𝑉Set[𝐴] \) of \( 𝐴 \), we use the corresponding value \( 𝑍 \) of \( 𝑌 \) to replace \( 𝑉Set[𝐴] \). This form of suppression can be effectively applied to the entire microdata DB (and in this sense it is "global") and is inherently recursive as multiple hierarchical roll-ups may be needed to guarantee anonymity.

### 4.4 Enhancing Anonymization

We conclude the section by discussing two advanced topics: embedding of complex business knowledge, where we showcase the use of domain experience for context aware anonymization, and implementation of runtime heuristics, to maximize the statistical effectiveness of our approach.

**Embedding complex business knowledge.** The overall anonymization process can largely benefit from the surrounding business knowledge, an aspect often neglected by dedicated tools. Thanks to reasoning, we can inject business representations into different phases of Algorithm 2: in risk estimation modules, to craft ad-hoc methods; into anonymization techniques, e.g., to opt for specific values for global recoding, and so on. The setting we show here, motivated by our experience in the Bank of Italy with financial networks, consists of taking into account the relationships that exist between the respondents, say \( 𝑋 \) and \( 𝑌 \). It is in fact common that the statistical disclosure risk propagates along linked entities, e.g., companies or people, so that being able to re-identify one, makes it easier to re-identify others. In essence, all the linked entities of a given cluster, have the same disclosure risk, obtained as the probability that at least one entity of the cluster is re-identified: \( 1 - \prod_{𝑐} (1 - 𝜌_𝑐) \), where \( 𝜌_𝑐 \) is the risk of an entity, calculated with one of the techniques in Section 4.2. Here, along the lines of what usually done to estimate the risk of households and hierarchical structures [26], re-identification risk is interpreted as the re-identification probability.

Now, types of links that can be considered are arbitrarily complex: finding members of the same family, companies of the same company group are examples. The latter, e.g., could be encoded by the following VADALOG rules: (1) Own(X, Y, W), \( W > 0.5 \rightarrow \) rel(X, Y). (2) rel(X, Z), Own(Z, Y, W), msum(W, (Z)) > 0.5 \( \rightarrow \) rel(X, Y). Clusters of companies (rel(X, Y) holds where \( 𝑋 \) and \( 𝑌 \) are in the same cluster) are defined by company control relationships: if \( 𝑋 \) owns more than 50% of the shares of \( 𝑌 \) (Rule 1) or controls a set of companies \( 𝐽 \) that jointly own more than 50% of \( 𝑌 \), than \( 𝑋 \) controls \( 𝑌 \) and thus \( 𝑋 \) and \( 𝑌 \) are in the same cluster.

Algorithm 9 shows an enhanced version of Algorithm 2, where the risk for a tuple \( 𝐼_2 \) is estimated as explained. Specifically, Rule 2 uses \#rel (and we assume here rel(X, Y) holds) to compute the risk for \( 𝐼_1 \) as the combined risk of the entities in the same cluster. The aggregation mprod is the monotonic product, which considers, for each contributor \( 𝐼_2 \), the maximum contribution it provides, so as to account for the new less risky anonymized tuples produced by Rule 3, and eventually triggering Rule 4.

**Runtime heuristics.** We have seen how VADA-SA operates incrementally and applies anonymization steps, only when tuples exhibit an overly high statistical disclosure risk. However, there are still various open questions to be addressed, which correspond to specific degrees of freedom in anonymizing microdata. If there are two or more tuples that violate the risk threshold, which ones should be anonymized first? Moreover, if there are two or more quasi-identifiers of the same tuple, which one should be suppressed or recoded first?

As for the first question, in VADA-SA, we adopt a greedy approach and choose to anonymize first the tuples that carry less statistical significance (namely, data utility), which can be estimated on the basis of the sampling weight. We exploit the so-called routing strategies [5] of the underlying VADALOG system to decide which bindings of the rule body to privilege when multiple possibilities arise. The approach here is quite intuitive: a "less significant first" strategy sorts the bindings of Rule 2 by risk and guides the anonymization accordingly.

The second question, namely the prioritization of quasi-identifiers, requires more care. We have seen in Algorithms 7 and 8 that either an existential or a higher-level domain value is used to replace quasi-identifiers and the specific attribute to consider is chosen as a consequence of the binding of Cat(𝑀, 𝐴, Quasi-identifier). Also in this case, we can prioritize bindings by adopting a VADALOG routing strategy and the greedy approach. In particular, a "most risky first" strategy would first bind the rules against the attributes that affect more the tuple-level disclosure risk. So, in this case, the strategy itself would rely on a VADALOG program computing the risk, in order to take informed decisions. For instance, consider the problem of anonymizing tuple 1 of Figure 5a. Applying local suppression on Sector removes any sample unique of the tuple, which then occurs with frequency 5; instead, applying local suppression on Area, e.g., would leave the value "Textiles" for Sector, and would then require further local suppressions until such attribute is removed, with a consequential loss of data utility. In other terms, a greedy approach to local suppression or global recoding sustains the preservation of data utility.

5 **EXPERIMENTS**

VADA-SA has been fully implemented and engineered in the VADALOG System. Towards a production application of the framework for the Research Data Center of the Bank of Italy, the system has been extensively experimented on real-world datasets from the Bank of Italy and synthetic ones to assess its anonymization capability (Section 5.1) and scalability (Section 5.2). The schema independent approach makes the framework general purpose and suitable for treating datasets in any domain.

**Datasets.** The microdata DBs used in the experimental analysis are reported in Figure 6. The real-world and realistic datasets derive from the Inflation and Growth Survey of the Bank of Italy, whose schema is shown in Figure 1. The synthetic datasets have been generated by fitting the real-world distribution (denoted by "W" in the figure) or by inducing specific unbalanced or very unbalanced distributions (denoted by "U" and "V"). Unbalanced
5.1 Testing Anonymization Capability

We analyzed the capability of the system to detect tuples that need to be anonymized (i.e., the "risky" tuples).

**Reduction of risk vs. loss of information.** We applied the Vada-Sa anonymization cycle to the real-world dataset R25A4W with the k-anonymity risk evaluation technique (Section 4.2) and choosing a risk threshold $T = 0.5$. We employed local suppression anonymization (Section 4.3), with a less significant first runtime heuristic (Section 4.4). We varied the anonymity threshold $k$ from 2 to 5. We adopted two metrics to evaluate the capability of the system to detect risky tuples: we counted the number of nulls injected by the local suppression as a result of risk evaluation, so analyzing how many values the system was able to erase (Figure 7a); we estimated the loss of information by weighing the number of erased values (i.e., the injected nulls) by the maximum total number of values, those of quasi-identifiers of the risky tuples w.r.t. $T$, that can be theoretically removed (Figure 7b) to satisfy the k-anonymity requirement. For both the measurements, we also evaluated the robustness of the approach, by applying the anonymization cycle to artificial but realistic datasets (R25A4U and R25A4V) having the same distribution of quasi-identifiers of R25A4W, but with an increased number of risky tuples.

The results in Figure 7a confirm what expected: when the k-anonymity threshold is increased, the anonymization cycle becomes less tolerant, and more redundancy is required to guarantee anonymity, therefore, in absolute terms, more and more labelled nulls need to be added to suppress specific values. We also observe that the number of needed nulls linearly grows with the tolerance threshold, as a consequence of an overall uniform distribution of values in the adopted combination of 4 quasi-identifiers. While an average real-world dataset requires less than 50 labelled nulls for 25k tuples with a 5-tuples tolerance threshold, more unbalanced versions require more, while confirming the trend. Figure 7b witnesses very good behaviour of Vada-Sa in terms of statistical preservation. For the real-world and the mildly unbalanced dataset, the loss of information is constantly below 20%, in particular between 12% (lower bound of R25A4W) and 17% (upper bound of R25A4U). For these two datasets, the constant trends show that when the number of risky tuples increases, the greedy approach succeeds in removing the values with a wider risk reduction effect. The loss of information for the very unbalanced dataset R25A4V is clearly higher, 37%, but it interestingly drops to 13% with less tolerant runs, because the high number of tuples that are considered risky on different combinations of values of quasi-identifiers, collapse in the k-anonymity comparison, when labelled nulls are introduced: so, while the number of nulls is high in absolute terms, the loss of information decreases. This result turns out to be an extremely positive guarantee of the anonymization capability of Vada-SA.

**Maybe-matching labelled nulls.** In this experiment, we want to assess the effectiveness of the maybe-match semantics, which we use to compare labelled nulls with one another (and has been described in Section 4.3), as opposed to the standard semantics of labelled nulls, such as that adopted in chase-based procedures (e.g., Skolem chase [11]). According to the standard semantics, for a quasi-identifier $qi$, we have that $q_{i} \neq q'_{i}$ holds if (i) $q_{i}$ and $q'_{i}$ have the same constant value, or (ii) both $q_{i}$ and $q'_{i}$ are labelled with the same null symbol. We plugged this semantics into Vada-Sa, used the same real-world and realistic datasets of Figure 7b and report the number of injected nulls by k-anonymity threshold in Figure 7c. Also here, the risk threshold $T = 0.5$ has been used. The figure highlights the proliferation of symbols (the red lines) that takes place with the standard semantics, which is in fact unusable in this setting. By contrast, the probabilistic interpretation of nulls we foster, minimizes the number of labelled nulls (the light-blue lines, whose zoomed version is in Figure 7a).

**Using business knowledge.** We show the results of anonymization in a real-world setting where anonymization cycle is complemented with a set of VADALOG rules that produce derived extensional knowledge about control relationships between companies. The rules and the setting have been presented in detail in Section 4.4. For the test, we adopt the real-world dataset R25A4W and its tweaked unbalanced versions, R25A4U and R25A4V. We anonymize each of the datasets by estimating the risk with k-anonymity with $k = 2$ and $T = 0.5$. We measure the number of nulls injected by local suppression in 5 settings, with increasing number of inferred control relationships, from 0 to 400.

The results are shown in Figure 7d. With all the datasets, the number of injected nulls grows with the number of relationships between entities, which induce bigger and more risky clusters. The three distributions of the quasi-identifier values differently interact with the derived relationships: the more unbalanced the dataset is, the more tuples will be affected by the propagation of risk of the outliers, resulting into a globally risky dataset, to be severely anonymized. In real-world tests, relationships disclose many cases that deserve anonymization (from 9 in the case of 100 relationships to 38 for 400), while the propagation effect is maximized in the R25A4V dataset with an upper bound of 323 injected nulls for 300 relationships.

5.2 Testing Scalability

Given the characteristics of the data at hand to be anonymized, we need to make sure that our approach scales well. Although the anonymization cycle, risk estimation and anonymization of Vada-Sa are expressed in VADALOG, where reasoning is PTIME in data complexity [6], here we want to investigate on the specific runtime of the system in different settings.

**By dataset size.** We tested the scalability of Vada-Sa by increasing volumes, with 4 synthetic datasets (from R6A4U to R100A4U), unbalanced and having a high number of risky tuples. We measured the elapsed time for the entire anonymization cycle and also pointed out the sole risk estimation component, with 3 different risk estimation techniques ([individual risk, k-anonymity, SUDA]). We used $k = 2$ for k-anonymity, $3$ as the MSU threshold.
for SUDA (see Section 4.3) and $T = 0.5$. We executed each run 5 times and averaged the measurements, for a total of 60 runs.

The results are shown in Figure 7e. All the three groups of trends confirm that the risk estimation component (dotted lines) dominate the elapsed time. This is reasonably expected, as the convergence of our anonymization cycle depends on a positive evaluation of risk estimation, which is then the bottleneck. The linear trend confirms the applicability of the approach. In particular, k-anonymity exhibits a very good behaviour, with elapsed time between 6 and 192 seconds for 100k tuples. The limited cost of estimation can be ascribed to the adoption of monotonic aggregations, which adopt incremental updates and need not be recomputed from time to time. Whilst in Algorithm 5 we have made a simple assumption to estimate the risk from the posterior distribution $F_k \mid f^q$ (which would have led to elapsed times similar to those for k-anonymity), for this experiment we plugged into Vada-Sa an off-the-shelf statistical library and sampled from the actual negative binomial distribution. The costly trend is motivated by the interaction overhead between the native VADALOG component and the library. The trend for SUDA is less than linear, with, e.g., 727 seconds for 50k and 1344 seconds for 100k tuples since the potential blowup on the number of examined combinations of quasi-identifiers is controlled by the VADALOG optimizations.

By number of quasi-identifiers. To investigate more the dependence of performance on the number of quasi-identifiers, we stressed Vada-Sa by anonymizing 6 datasets R50A4W-R50A49W, so with increasing number of attributes and fixed number of tuples, 50k, and real-world-like distribution. We used the same thresholds for k-anonymity, SUDA, and $T$. We measured elapsed time and the risk estimation component. We executed each run 5 times averaging the results, for a total of 90 runs.

Figure 7f reports the results. As expected, individual risk and k-anonymity are only marginally affected by the increased number of quasi-identifiers, as they do not consider all the combinations with at most $k$ attributes, but only those with exactly $k$. Instead, we may expect a much worse trend for SUDA, where for each tuple, all the combinations of at most $k$ attributes are inspected to detect potential MSU. Remarkably, no combinatorial blowup appears in the figure, witnessing a very effective behaviour of the VADALOG execution optimization: while the activation of Rules 2-5 of Algorithm 6 could in theory cause a blowup w.r.t. $k$, it does not happen in practice because the greedy activation of Rule 7 performed by VADALOG to detect the MSUs preempts the generation of redundant combinations of quasi-identifiers.

6 RELATED WORK

Statistical disclosure control is a broad topic to which many have contributed, especially from the Statistics community, whose work can be considered related to ours.

The concept of Sample Uniqueness (SU) to measuring the risk of data disclosure was introduced by Skinner [35], while k-anonymity, was first presented by Sweeney [37], along with the first methods of anonymization by generalization (our global recoding) and local suppression. The measure of individual risk in our contribution is inspired by the work of Benedetti and Francconi [7] who proposed to compute the risk of data disclosure with the sampling weights of data records.

The topic of data anonymization is related to the area of differential privacy [17], where an interesting concept may be adopted in our approach so as to develop a new family of risk measures.
based on the idea that an individual’s privacy may be violated even knowing the absence of the individual from the microdata. Investigating such direction will be matter of future work.

While the foundations of our work are set in the theory of statistical disclosure control, our contribution is concerned with providing an industrial production ready solution for the Bank of Italy, conveying a set of properties that derive from a fully declarative reasoning approach. In this sense, we combine our experience in logic-based reasoning [6] and schema-independent solutions to model management problems [3]. None of the existing dedicated software solutions for statistical disclosure control offers the mentioned set of properties. The software pack AR-GUS [27] aims at local suppression and coding, as does the Datality system [36]. Manning et al. introduce SUDA2 (Special Unique Detection Algorithm) [29], whose objective is to detect the risk in certain unique combinations of variables. Recently, the R package sdcMicro has implemented many of the risk measures and anonymization approaches of our interest [9]. Likewise, ARX is a solution for data anonymization that has been proposed as a practical approach to Statistical disclosure control [33]. A comprehensive survey of the statistical approaches has been provided by Matthews and Harel [30]. Recent work on the risk of information disclosure in linked data, and, more in general, ontology-based data, has formalized the problem and defined its logical foundations [8], with an interest in the concept of linkage safety in RDF graphs [24]; a declarative framework for linked data anonymization has also been proposed [15]. The problem of preserving privacy in data exchange has been analyzed also in the context of information integration systems [31] where a practical solution is represented by MapRepair [10] and in the cryptography community, with homomorphic encryption [28].

In the AI literature, statistical disclosure control has been mostly considered within machine learning [16] and deep learning approaches [4]. Yet, they have a different focus and aim at generating anonymized clones of existing datasets while respecting the original statistical properties. An interesting deductive proposal by Øhrn and Ohno-Machado uses Boolean reasoning for data anonymization in databases [32], which however remains purely theoretical and just considers the combinatorial aspect.

7 CONCLUSION

In this paper, we presented VADA-SA, a declarative statistical disclosure control framework. We demonstrated the anonymization workflow, metadata dictionary, and statistical disclosure risk estimation. Utilizing these components, we introduced the anonymization cycle. To maximize the statistical effectiveness of our approach, we also presented two enhancements, namely embedding of complex business knowledge and runtime heuristics. We validated the approach on real-world central bank data. As future work, we plan to further enhance the framework, and test it in a variety of other real-world scenarios.

REFERENCES

[23] Georg Gottlob, Thomas Lukasiewicz, and Andreas Perras. 2014. Datalog+/: Questions and Answers In KR.