Selim Soufargi

Privacy-Preserving Multidimensional Big Data Analytics over Big Data Lakes: Models, Techniques, Algorithms

May 15, 2024

Abstract

It is well established that health information privacy is of crucial value and importance to the good execution of analytical processes. The role of analytics, on the other hand, is also known to be critical for precision medicine as well as for accurate healthcare recommendations. While data analytics try to uncover patterns in the underlying health data to support decision making, privacy, priorly, ensures that these data are not exposing sensitive information about the individuals on which the analytical process is being applied. Unfortunately, existing research works usually put emphasis on solving either data privacy or data analytics issues in such a way that leads to poor analytical outcomes in terms of accuracy for decision making support. This challenge motivates our need for a joint paradigm between data privacy and data analytics to enhance the capabilities of current analysis of healthcare data to eventually enable precision medicine. Indeed, a joint paradigm would involve the creation of algorithms that consider a fully integrated process that enables data analytics while preventing the disclosure of identity information. In this thesis, we propose several algorithms, frameworks and techniques that specifically address the previous matters and challenges in data lake contexts. Indeed, we aim at developing privacy-preserving data analytics techniques in big data lake environments based on different kinds of data types and settings. In fact, depending on the goal of the privacy preserving analytical tasks, we propose tailored frameworks that, we argue, effectively and efficiently support the creation of health related recommendations and thus, in the QUALITOP context, support quality of life after treatments for cancer patients.

To my parents, for their love.

Contents

List of Tables

List of Figures

1

Introduction

Nowadays, Big Data are being generated from innumerable sources, whether they are IoT devices, social networks or more broadly cloud systems, the growth in volume of these data is real. The velocity at which these data are created is inprecedently elevated, and the variety of these data (type and format) are highly diverse. It goes without saying that such growth in data generation could lead to issues related to the privacy of data owners when these data are used to generate insights. Indeed, since data mining techniques have become much more complex, this could lead to increased security issues and expose data more than ever to disclosure [179]. In this sense, it is important that the trend in complexity and advancement in data mining and data analytics techniques is followed by and similar trend in privacy-preservation. In this thesis, we aim at developing privacy-preserving data analytics techniques in big data lake environments based on different kind of data types and settings.

1.1 Context

Quality of life (QoL) after immunotherapy treatment of cancer patients is the main focus of the EU H2020 QUALITOP project. The assumption that immunotherapy is a promising treatment for many types of cancer heavily influenced the project proposal. Its efficacity against melanoma (up to 60% objective response rate) as well as against acute lymphoblastic leukaemia (80% complete response rate) motivated the current project proposal which targets the evaluation and the assesment of the QoL in patients having been treated with immunotherapy as well as the monitoring of their health status along the period of the treatment. More specifically, the project goal is to define "predictive markers" of occurrence of immunotherapy-related adverse events so that a comprehensive approach could be devised on-time in order to promote their QoL. From our perspective, we aim to devise analytical methods to help promoting the QoL in cancer patients treated with immunotherapy.

In order to accomplish the mentioned goals, we faced numerous challenges that needed to be addressed in proper ways as to circumvent the hindering obstacles related mainly to the 5Vs of big data.

Indeed, in an era where data are available more than ever in every type and format, numerous attempts at deriving valuable insights from data for a specific purpose have seen the light of day. Indeed, Big data systems, techniques and frameworks based on cloud computing and leveraging cloud storage resources have considerably alleviated many issues related to data analysis already. Yet, more effective systems/algorithms could be developed based on the specific needs of the analysis. In the current research work, we aim to define and implement solutions that are tailored to the data analysis requirements and that fit into the aforementioned project goals.

1.2 Privacy-Preserving Big Data: Motivations, Goals and Examples

With the advent of Big data mostly in the last decade or so, data have leveled up when it comes to format, size and speed of generation: Indeed, data are more than ever diverse (different type and format: 5V's Variety), are of bigger size (5V's Volume) and generated at a faster pace (through enabling technologies: 5V's Velocity). The big data processing area comes into play to enable large scale analysis of such data and the generation of valuable insights. For example, in healthcare area, data are nowadays on an increasing slope when it comes to their generated volume whichl evolves at the staggering order of the petabytes or even yottabytes [100], [23]. Another example is social big data, where volumes are drastically increasing and they especially did in the last decade [26]

With this massive unpredicted cumulation of data, a paradigm shift imposes itself on big data computing processes that yields analytics needed to comprehend these data and make use of them for diagnostic or predictive purposes. Yet, an important aspect to be considered before the process of generating analytics is the data privacy of such high volumes of amassed data.

In this sense, and given that data are very sensitive and that different countries consider that, for example, healthcare data as legally possessed by the patients [195], protecting these data must ensured by the sharing party.

In fact, the protection of data has become a fundamental issue and many regulatory frameworks have been defined to protect individual data. Examples are the European Union's (EU) General Data Protection Regulation (GDPR), and the United States of America's Health Insurance Portability and Accountability Act (HIPAA).

A full data sharing process in the healthcare domain would be as follows: a patient (record owner) who's provided their information to a hospital (data publisher) for medical diagnosis purposes wouldn't like to see their information

disclosed to a third party (data recipient) which would use their information for data analysis purposes. As shown in Figure 1.1, data publishing requires that the data publisher ensures the anonymization of the patient data before sharing them with a data recipient. Another pertinent example would be the risk of spatial information exposure of social media users [14].

Furthermore, and in order to enable data privacy, many frameworks could help in achieving such goal. In fact, data privacy is many-fold and could be achieved through different paradigms or processes separately or jointly adopted. For example federated databases emanating from the growth, in diversity, of Big Data are an effective solution for data privacy that primarly ensure data locality which enforces privacy-preservation. On the other hand, data privacy of such system could be enforced by using tailored state of the art anonymization algorithms.

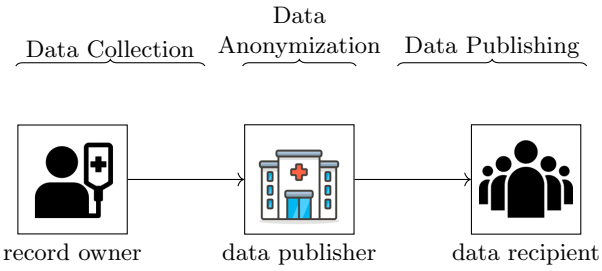


Fig. 1.1: Privacy-Preserving Data Publishing Process

1.3 Big data Analytics

Whether they are tailored for big data warehouses in the case of multi-dimensionality or more broadly are Cloud-based, a soaring number of literature works related to extracting big data analytics from a wide variety of data sources has characterized the last decade. In the next paragraphs, we describe briefly these two mentioned environments while describing the most prominent research done in these topics.

1.3.1 Data Warehouses

Are highly structured databases where multi-dimensional data are stored for serving specific analytical purposes based on domains (Data marts). Indeed, data marts business/topic focused databases that serve specific analytical goals. The Typical data loading process into data warehouses is based on Extract (from the data sources) Transform (to meet the stucture of the data warehouse) and Load (into the data warehouse storage) (ETL) based on a

schema-on-write approach (defining the schema of the data at the time of loading). The end goal of such systems is to enable querying and visualization capabilities to meet Business Intelligence needs. Unfortunately these systems have limitations as these systems require highly structured data as input or also require larger computing and storing resources for dealing with larger datasets. Literature detailing the architecture of such systems are [185, 18]

1.3.2 Big Data Processing on the Cloud

Cloud-based data processing systems come at rescue to replace traditional data warehouses for big data analysis by leveraging both computing and storage cloud capabilities. Also, given the variety of data nowadays available, these new systems propose to deal with not only structured data but also semi-structured and unstructured types of data. A maturing solution are the data lakes. The literature has been struggling to reach a consensus on what would be define data lakes. The earliest works have argued that a data lake is a low cost data storage system [88] which is tightly coupled with Apache Hadoop that would define its main core. The same work emphasises how a data lake should enable data diversity (various data types and formats), should leverage a schema-on-read approach (the schema of the data is unknown at ingestion time but only at processing time) and enable data provenance (disclosing information such as the data source, data versioning...) They also highlight the fact that tailored analysis tools should be exploited.

On the other hand [107] advocates the use of data lakes as information systems which entails that the former embeds capabilities to deal with, not only raw data, but also highly-structured data such as the case of OLAP cubes data in data warehouses. The data cubes will then be filled using an ETL process that is based on the data inside the data lake. The authors also pinpoint the importance of leveraging metadata and user roles (leading to better data governance) across the data lake. The work demonstrates why it is important for a data lake to emcompasse three modes of data aquisition namely: batch, real time (streaming) and hybrid.

For how we see a proper definition of what is a data lake, we would rather put it this way:

A data lake is a centralized or a distributed repository that enables the storage of a wide plethora of data types to be persisted for analysis purposes at scale. At storing time, metadata availability shouldn't be an obstacle for identifying and filtering the data for the analysis as long as attribute names and related types are defined.

Indeed a centralized (or monolithic) data lake would store data as-is coming from several sources into one sole storage system in order to enable a schema-on-read approach for their analysis. Contrastly, a de-centralized (or distributed) data lake (also referred as data mesh) enables the storage of datasets ingested by domain (or type) and enables the easy proliferation of data sources leading mainly to an easier scaling of the data lake. Data lakes

should also be able to define data lineage and provenance mechanism to avoid themselves becoming "data swamps". Numerous works have also defined a proper architecture for a data lake, examples are elaborated in [42, 17, 16, 15]. Furthermore, data lakehouses are designed to deal with multi-dimensional data in addition to the potential unstructured data within data lakes. In fact, data lakehouses are fusioned data lakes and data warehouses. As a benchmark data lakehouse architecture, we choose the one depicted in Figure 1.2.

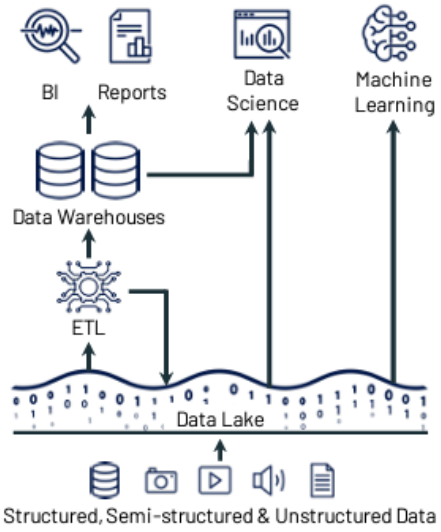


Fig. 1.2: Reference Data lakehouse [212]

Indeed, in this specific setting, data analytics are derived using data science techniques including machine learning from the data lake and also Extract, Transform and Load method can be applied on the data in the lake to create a data warehouse from which multi-dimensional data could also be analyzed. In this sense, we aim at defining the proper tools, methods and algorithms that support building such system as a data lakehouse while emphasising on the analytical and privacy aspects.

1.3.3 Big Data Analytics Categories

Big data analytics play a crucial role in guiding decision making across enterprises and organization. Indeed, data analytics provide knowledge of value that positively impact and direct decision making by enlightening about hidden insights within the data. This leads us to strongly affirm that big data

analytics support decision making, an affirmation consolidated and confirmed by many research works as in [13, 122, 87, 89, 168]. In the following, we delve into the intrinsics of data analytics, by categorizing them into four known types namely: descriptive, diagnostic, predictive and prescriptive types of data analytics. In order to distinguish between these analytics Figure 1.3 depicts the information value in function of the complexity in decision making: specifically the data input should be of high value in case the decision making aim is hindsight or insight since we derive these directly from the data themselves and of less value if we automate the decision making process (making it much more complex e.g., using machine learning) in which case the realized value of the process would be much higher (predictive or prescriptive).

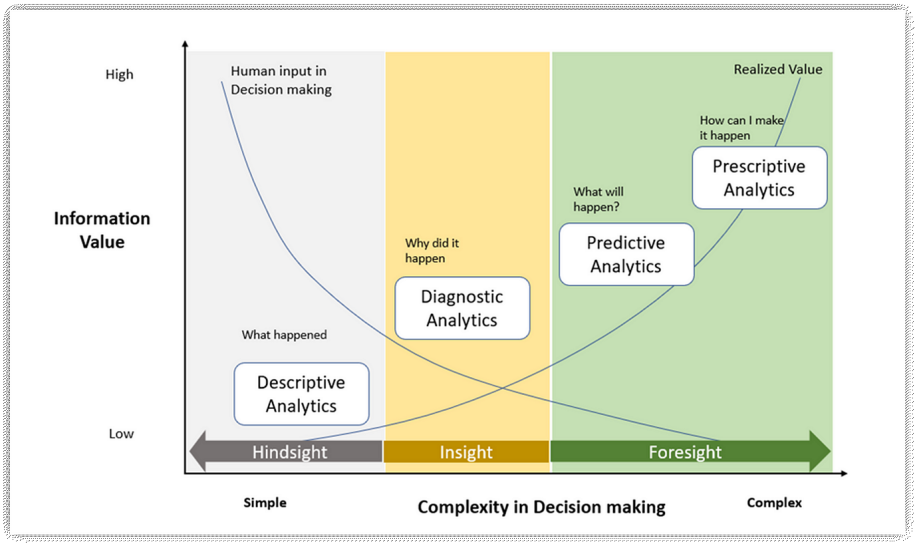


Fig. 1.3: Gartner's Analytics Model [1]

Descriptive Analytics

Descriptive analytics summarize the data at-hand and identify trends and patterns through attempting to answer the question "what has happened?" that is what is happening and what happened. These analytics are informative on the current state of things useful as insights for the future. Numerous statistical measures could be used to draw conclusions out of the data. For example, we mention, measures of variability (expressed throiugh variance) or measures of divergence from normality (expressed through standard deviation). Other distribution related informative measures are the mean and the median measures. The work [199] defines this type of analytics as a summa-

rized form of data description from which insights could be concluded as to answer the related mentioned question.

Diagnostic Analytics

Diagnostic analytics aim at identifying key factors causing specific observations. These analytics goal is to answer the question of "why did it happen?". Eventually, diagnostic analysis will help unveiling the hidden patterns in the observed events and will help determining the cause of trends and possibly the correlations between the data features.

Predictive Analytics

Based on actual data, predictive analytics address the question "what will happen and why?" to uncover the future and make assumptions on possible outcomes based on the past data. In other words, predictive analytics aim to make predictions on various events based on the seen observations. In short, predictive models employ statistical functions to analyze past and current observations for predictive purposes. According to [121], this type of analytics help in spotting future event patterns to enable predictions over unseen observations.

Prescriptive Analytics

Prescriptive analytics aim at drawing "the full picture" of the seen observations as to suggest actions in favor of the goal of the analysis. This type of analytics answers the question "what and shall we do and for which reason?". This type of analytics help in determining what action should be performed in light of to what happened. As a matter of fact, prescriptive analytics can leverage both descriptive and predictive analytics in order to yield business decisions [44].

1.4 Research Contributions

In this thesis, the main contributions are three-fold:

- A novel privacy-preserving framework is defined, implemented and evaluated across numerous performance and privacy metrics where the data are made private through state of the art processes while diversity of minority values across data domains are guaranteed. The framework is specially suitable for data sharing in data lakes environment since it proposes a tree-like data model enabling the easy search of anonymized data.

- Analytics generation problem is tackled considering real-life as well as synthetic data in distributed environments. Real use cases were developed on such data using the implemented federated system. More specifically a full-fledged cloud system is designed and implemented to mimic a distributed-storage type of datalake where QUALITOP data could be analyzed. The system is equipped with an analytical dashboard useful for descriptive and predictive purposes. The implemented system leverages state-of-the-art big data storage and processing solutions such as Apache Hadoop, Apache Yarn, Apache Spark or also Apache Hive.
- An analytical methodology based on disparate multiple structured data (for data warehouse solutions) is created and tested over real use cases. Indeed, cloud data warehousing is an integral part of a fully-developed data lake, based on this assumption we derive analytics from distributed datasets using an innovative OLAP technique.

1.5 Overview of thesis structure

This thesis is organized as follows:

Chapter 2 presents a summary of related works of our contributions made throughouht this thesis, described through chapter 4 to chapter 6.

Chapter 3 presents a literature overview of privacy-preserving frameworks, data analytics systems and OLAP analytical techniques. We address the most important algorithms, tools and frameworks that are the closest to our proposed works across this thesis. This chapter is devoted to showing how our works fit into the literature ergo preparing the ground for a comparison of our works with literature proposals in privacy-preserving big data analytics.

Chapter 4 presents QFLS: a complex federated learning system that enables the analysis of large scale data in a distributed environment. We tackle data issues such as scalability, security in order to provide a system capable of managing diverse sources of data in a privacy preserving manner (mainly through preserving data locality). To assess its potential we have created use cases based on real life datasets to show how effective and efficient it is to deal with such data in a distributed environment. We also proposed a data analytical model that copes with hierarchical data to provide meaningful insights over the target data and eventually provide predictive analytics helpful for practitioners and medical doctors to provide recommendations for the cancer patients.

Chapter 5 is about defining and assessing a privacy preserving framework that deals with hierarchical healthcare data. The framework is designed to protect sensitive datasets at analytical time while enabling precision medicine primarly by preserving the minority groups. A proper cloud based assessment of the algorithm is provided and a user-interface is offered to facilitate the execution of the algorithms over the input data.

In Chapter 6, we seek to apply the same purpose of our thesis in a different context that of OLAP. We define and evaluate an OLAP based privacy preserving technique that deals with analyzing OLAP cubes. As shown through the experiments conducted over synthetic cubes, our technique is able to derive meaningful analytics over the input data, namely, correlation based analytics.

Finally, Chapter 7 presents the conclusions of this thesis while summarizing the main contributions made.

2

Related Work

In the first chapter, we have defined different topics that from near or far link to our current thesis proposed techniques, we have also introduced the context of the work and the technical environments constraints and goals that this thesis is about. In this chapter we focus on describing the related works that relate to our main contributions we made in this thesis mainly for what regards the privacy-preserving multidimensional big data analytics realm. Whether it may be in data lake contexts (Section 2.2), or in the distributed environment context specific to healthcare data (Section 2.1), or in OLAP contexts (Section 2.3), current solutions have the tendency to focus too much on only one aspect/goal without considering other potentially analytics hampering aspects. Those aspects usually improve the analytical outcome and if sufficiently flexible could lead to precision medicine.

2.1 Federated Big Data Analytics Learning Systems in the Healthcare Area

Many literature proposals have been defining a plethora of solutions to deal with data privacy in systems aiming at deriving analytics from all sorts of data (batch or stream) within different settings and environments (centralized or distributed). By definition, federated systems aim at both data insight generation and at the same time data privacy. Because data locality is preserved in such settings, data privacy is enforced. In what follows, we review and discuss the most critical and effective frameworks and systems that leverage of data federation. While some of the works enable cross-organization data analytics while ensuring data are safe involving third party members, others approach data analytics generation through data anonymity techniques such as homomorphic encryption, differential privacy or simply k-anonymity. In a federated settting, [180] research work goal is to integrate the Institutional Review Boards (IRB) privacy authority into a healthcare data query platform in order to re-enforce privacy for data sharing among investigators. As

a matter of fact, IRBs role is to ensure that research of human subject nature is regulated as what concerns data usage and access from third party researcher organizations. Treatment and prevention from human diseases requires leveraging human based real data that must be protected from illegal accesses and eventual fraudulent use. In this sense, the mentioned work attempts at designing and implementing an integrated federated platform that ensures that IRBs allow data access for an investigator to use patient data. Hitherto the current proposal, IRBs denial or acceptance of data use had to go through a tedious paperwork-based process. The current work, then, tries to automitize the IRBs process and integrate it in a federated system. The integrated system features a web-interface an associated IRBs (or several IRBs) based on data use contracts to permit or reject researchers to query data, and allow them to do so only based on strict data access rules issued by IRB reviews.The access rules would restrict the data access to researchers for the strictly the data they would need to accomplish their analysis needs. Other privacy mechanisms mentioned in this work are, federated authentication and authorization, digital signatures and message encryption.

On the other hand, in [200] an aggregator-adapter model is defined and implemented through the proposed system named *SHRINE*. SHRINE is composed of a Query Aggregator (with front-end interface and web-services based backend) that interacts by means of AJAX with locally distant Adapters located at hospitals and receiving user submitted queries from the user-interface and returning patient counts. For simplicity reasons, the main filtering features of the patients for the queries were demographics based features as well as diagnosis ones and the authors estimated that a common ontology for all SHRINE databases is a suitable solution but time and effort consuming and is costly to realize.

An example of federated systems emphasizing less on the privacy aspect of data and more on the heterogeneous data management aspect is described in [110]. The system is composed of a set of clusters (master and slaves) which is able to process the heterogeneous data through a series of Map-Reduce computations executed over each of the heterogeneous sources of data. The typical process is as follows: data are processed through a Merge phase into a list of useful insights using the reduced partial results. The Map-Reduce-Merge tasks are performed thanks to the regional clusters where the heterogeneous data sources are located. A global-Reduce task, executed on a master cluster datanode, enables a global reduction operation over the gathered sub-results obtained from the regional clusters thus yielding one final output. To reach the regional clusters, user-defined queries are splitted into sub-tasks (according to the nodes availability and capacity and the dataset distribution) and then assigned to a regional cluster for execution.

In [45], authors emphasis more on the data integration aspect of data federation and propose a system's architecture that leverages federated query execution onto a variety of heterogeneous data sources such as relational data, web data, social and deep web to serve a data search purposes. A query builder

component is responsible for converting the user-defined query string into a suitable type of query for the target data source (e.g., query strings may be converted in SPARQL syntax for RDF-based data sources, or also into SQL syntax for the case of relational data sources). All returned query answers are transformed into a RDF-based graph holding the results. In addition, all RDF-based local results are aggregated according to a global schema approach into an RDF-based global result graph. The final results in the user-interface are conveniently displayed in JSON-LD format.

Yet another research work [135] proposes a *k-anonymity* based type of queries to probe a variety of data sources in a distributed and scalable environment while emphasizing on the efficiency of the querying process. This type of private queries would protect against side-channels attacks that federated data querying suffers from. The proposed method, builds a direct acyclic graph (DAG) of database operators after parsing the SQL query. A query planner then transforms the query into k-anonymous query execution plan that will be forwarded to the data owners. Data owners will execute the query over the union of their datasets and will return an encrypted result to the client. The homorphic encryption (HE) at the service of data federation example comes next in [111] where a system architecture is detailed showing how they not only stopped at applying the cryptographic method on the local data (which results in big overhead) but they propose a more efficient way of using HE on the federated nodes data. Indeed, the systems features a selective parameter encryption function that guarantees that only highly sensitive data weights would be homomorphically encrypted and aggregated into the global model at the server: first privacy sensitivities are calculated for each local dataset, an encryption mask will then be generated based on the aggregated model privacy map. This process will make sure that only fewer weights will be encrypted at the global model ergo improving overheads. The algorithm at each node define a differential privacy component as an option, where local data could be seen added noise whenever required.

In addition to the previous examples of federated systems, a particular type of such systems is what is called data frabric (or data mesh) [75]. A data fabric/mesh is a federated system where data storages are scattered by data domain. This type of setting enables the data of many organizations to be analyzed through cloud capabilities independenltly of their type or size. In fact, through a data mesh, each data domain could leverage a different cloud computation resource that is each domain could be associated with one or multiple cloud platforms. It goes without saying that the proliferation of data origins is a much flexible process through data mesh systems and more liberty on the choice of the cloud platforms that would bring options related to security, reliability, cost constraints as well options related to visualuzation capabilties and other data related functions (that one cloud platform may offer and not ther other), and that based on each specific data.

In the following we list and describe some of the prominent works done in the field of privacy-preservation big data.

2.2 Privacy-Preserving Big Data Publishing in Big Data Lake Contexts

Most of literature works on privacy-preserving data sharing tackle the topic in either centralized or distributed settings, with or without applying machine learning processes and are based on the type of data in question. For example some works have anonymized IoT based data through integrating a blockchain approach to deny or allow data accesses, some others make use of machine learning models to generate a privacy preserving representation of the data. In a nutshell, most of these techniques, however diverse they could be, account for the large size of the input data thus could be useful in big data lake contexts.

In what follow we detail some of the closest works to *AB-DOM*: The work in [84] describes a healthcare platform that is showcased for two use cases namely arrhythmia and stress detections. The platform makes the data private for analytical methods employing AI (through neural networks or clustering) or through basic statistical methods (such as COUNT). After data are submitted to the papaya platform, AI based or basic statistics methods will be safely applied on data to derive analytics (e.g. whether the patient suffers from arrhytmia or stress).

The second work, although is more suitable in IoT environments uses an inventive approach to anonymize sensors gathered data. Indeed, in a body area network (BAN) context for connected-Health [43] defines a data privacy approach, suitable most for ECG data, that is based on differential privacy method incorporating a dynamic noise threshold mechanism suitable for supporting data analysis. An interference threshold (noise parameter) is defined for each feature according to a measured importance level. The noise is accordingly added to data when the differential privacy constraints are not satisfied. The differential privacy probability distribution is altered to obey to the statistical variance of the data.

An additional work done on IoT case is the work [20] which starts by highlighting the fact that neural networks model are subject to model inversion attacks then proposes an algorithm that trains privacy protected model as follows: sensitive attributes related data have their gradient perturbed using a differential privacy compliant method, while other attributes are less considered in the process. The authors demonstrate that their algorithm is applicable in a distributed setting, where models are trained in different locations, a disposition well-tailored for IoT devices and telemedicine systems. Experiments have shown that model accuracy as well as the attack accuracy are controllable through an input parameter (the attack accuracy could be shrinked to zero) and that the decay in attack accuracy is substantially greater than of the model accuracy.

Yet, another work described in [171], details about a blockchain based method relying on a dynamic approach to patient consent for third party data sharing suitable for analytics mining. A use case example is described

where a hospital requiring patient data to train and test a model for generating analytics would ask consent of to access patient data by the caregiver. The data access is based on a blockchain process where a token based access control is employed. Blockchain would mostly support traceability and and legal compliance ergo enforcing data privacy. Through this approach data analytics would be boosted given that hospitals access the full data whenever it is possible (through a deliberate patient consent) to process them and yield insights useful, for example, for medical recommendation.

Finally, [114] starts from noticing that privacy preserving techniques for data sharing cause distortion in healthcare data that reflects negatively on data utility. The research work presents a generative modelling framework that synthesizes data that are characterized by a high-fidelity (generated data are statistically the closest possible to the original data while keeping quasi-similar utility) and that are privacy-preserved. An encode-decoder solution is proposed in conjunction with a generative adversial network to generate the syntethic data: the encoder would receive the raw data and transform them into low-dimensional representations that would be fed to the GAN in the training phase. Then using any random vector they generate synthetic data through the GAN based generator (yielding synthetic representations) and the decoder (yielding actual synthetic data from the representations). Both encoder and decoder are trained using a cross-entropy objective function.

As we have seen, numerous works have been devoted to privacy-protect data in all sorts of domain especially in the healthcare data where patient information is considered sensitive in most part. For what concerns the context of big data lakes, data should be analyzed while the anonymity of the records should also be preserved as that represents a constraint of big importance for the data lakes' ingested data primarly because of the regulations previously mentioned.

In the following we list and describe a summary of works on privacy preserving multidimensional big data analytics topic.

2.3 Privacy-Preserving Multidimensional Big Data Analytics

Literature examples targeting the topic of privacy-preserving Multidimensional Big Data Analytics are almost innumerable. Yet, these differ in contexts they apply for: for example, some works are adequate for IoT/Cloud environments others are applicable for specific types of data (mobile data, stream data, sensors data etc.) only whereas some others are specific to certain type of systems (cyber-physical etc.). In what follows there is an enumeration of some works that we think are relevant to our *Drill-CODA* technique in the sense that these techniques derive analytics while preserve the privacy of the data along the process. In [162] A privacy-preserving stream analytics

system called *PRIVAPPROX* is proposed. The authors start by highlighting that combining privacy-preserving analytics and approximate computing could be useful to implement a system that is highly effective in privacy-preserving data while generating analytics from real-time stream data. The proposed system leverages the power of approximate computing (guaranteeing low-latency/efficient analytics in distributed environments) via sampling in combination with differential privacy (adding noise to source data) for enforced data privacy leading to leads us to achieve zero-knowledge privacy [92]. Indeed, the authors consider the aggregate results to be a breaching of data privacy in themselves, so they propose to use binary histogram buckets (for numerical data) where each bucket is a range in the query answer. If the result of a SQL query falls into the range values then the bucket will be set to 1 otherwise it will be 0. Moreover, regular matching expressions are used for categorical buckets. In addition to the previous privacy measures, the solution proposes *anonymity* and *unlinkability* mechanisms, to make the association of a query answer with a client unfeasable and to thwart attempts to attribute any pair of query requests or answers to one specific client in the distributed setting, respectively through transmitting answers using a proxy. The proxy communication of the query answers are based on an XOR-based encryption combined with source rewriting.

Another technique aiming at deriving analytics from geo-spatial data while preserving their privacy is described in [151], specifically, a privacy preserving multi-layered, modular, and scalable architecture named *OPAL* is proposed in form of loosely coupled micro-services and which ultimate goal is to enable effective data analytics over location/mobility data. Indeed, OPAL typical use case is to compute the population density of a certain area for any given time interval in a geo-localisation data privacy preserving manner. The architecture encompasses first a data management layer responsible for monitoring jobs (run in MapReduce fashion), and scheduling the job requests for computation. Second, a storage layer needed to persist the mobile phone data that are ingested in a pseudonymized form. Third, a computation layer, offering the analysis algorithms MapReduce execution capabilities to the platform. Finally, an endpoint layer is included in the architecture to provide the user with the adequate API needed to perform the queries over the mobile data. Furthermore, the density analysis algorithm is described thoroughly while emphasizing on its integrated privacy and analysis components. Indeed, the privacy method used in the density algorithm is based on differential privacy (using the Planar Laplace distribution) technique required to obfuscate single user locations in the considered mobility data and hence achieve geo-indistinguishability. The privacy component includes also methods to suppress data or to add noise into them. The analysis part includes the aggregation operations that the algorithm integrates and which the platform is capable of performing over the data, and which are mainly: count, sum and median. Finally, the platform scalability is studied and the performance reported as to demonstrate its flexibility in dealing with large volume of mobile data.

Yet, [119] tackles the privacy preservation of data for data analytics in the case of cyber-physical systems. The authors start from observing that cyber-physical data are generally not of normal distribution type and that the independent component analysis (ICA) technique would be a suitable method to adopt in that case. Indeed, authors highlight the fact that by applying projection-based transformation such as ICA, the data privacy would be guarateed and at the same time data utility would be kept at a satisfying level for data analytics processes. Authors pinpoint the usefulness of ICA in diminishing mutual information between variables of the observed data while demonstrating the preservation of the data utility of the transformed data. The ICA method aims at mapping the observed non-gaussian data and transforming them into independent set of features ergo limiting variables correlation which in turn prevents original sensitive information from being discovered. Finally they present an algorithm that leverages ICA in order to attain their final goal of privacy preserving data while keeping data utility at a reasonable level for data analysis.

Finally, in [98] a novel privacy-preserving IoT sensors data analytics framework is described. The authors depart from noticing that the classic distributed model training, based on transmitting locally trained models and needed to derive a global model, may still be exploited to expose sensitive data. Building on the previous assumption, the main proposed technique consists in a distributed learning technique that instead fuses local analytics, namely predictions, resulting from local training models on different IoT nodes into global analytics. Finally, a neural network dynamic layer-wise evolution algorithm is proposed for each node model training in order to optimize the generation of analytics for each local node data in order to enhance the global fused prediction accuracy. The idea being to adjust the model structure so that neither over-fitting or under-fitting can occur. The algorithm is based on a asynchronous stochastic gradient descent optimization function so that the model parameters sent to the coordinator node (in the cloud) are processed in an asynchronous way. The algorithm adapts the neural network at each node by updating its weights (only if they provide better accuracy) and changes the hidden layers structure (by adding other layers if needed) as to comply to a data size by a neurons count ratio based constraint.

2.4 Summary

In this chapter, we have surveyed state-of-the-art solutions for data protection applied to data lakes and distributed environments (e.g., IoT), in the following, we cite few of their limitations compared to the proposals we will make throughout this thesis:

Current solutions focus entirely on anonymizing data whether they be EHR, IoT sensors based, or in the field of smart grids, they introduce cryptographic solutions, query based privacy, privacy through machine learning,

blockchain or also data transformation based approaches such as slicing, but they almost none of the current techniques, as far as we have researched, introduced a second issue to consider processing data for another important aspect related to privacy. This also means that the privacy techniques introduced don't account for the effectiveness of the output for the analytical process. Given this limitation, the privacy techniques of the current literature don't foster analytical processes to yield the most effective and accurate analysis over the considered data. In the current thesis, we deal with this through our designed techniques. In other words, the coupling between the anonymization process and the analytical process is weak to inexistent in the current described methods. This means that, current literature simply anonymizes the data to make it private and not to also prepare it for a more meaningful and useful analysis.

The remainder of this thesis is devoted to bridge the gaps mentioned as to enable a more flexible analytical processes in terms of the analysis potential. Through the proposed methods in the current thesis, the analytical processes are more aware of the different constraints that the input data might introduce; mainly, the distributed aspect or the privacy for the analytical processes.

3

Privacy-Preserving Multidimensional Big Data Analytics Models, Methods and Techniques: A Comprehensive Survey

In this chapter we list and describe some of the prominent works on cloud based systems for privacy preserving data analytics. We also try to link the state of the art with our proposals as to highlight as much as possible our key contribution reported through the current thesis. In particular we bend over few emerging key topics related to our thesis. Namely, we focus on federated settings and how they can be leveraged to provide privacy preservation for data analytical processes as well as big data lakes related security all in the healthcare domain.

Multidimensional big data analytics topic dates back to the dawn of the datawarehousing and business intelligence era [108]. Over time, the analytical processes stemming originally from OLAP and business warehouses, have evolved into a more complex, machine learning based, processes such as regression, clustering or pivoting. In order to illustrate a typical workflow involving these techniques, Figure. 3.1 depicts a cloud-based model architecture suitable for healthcare data.

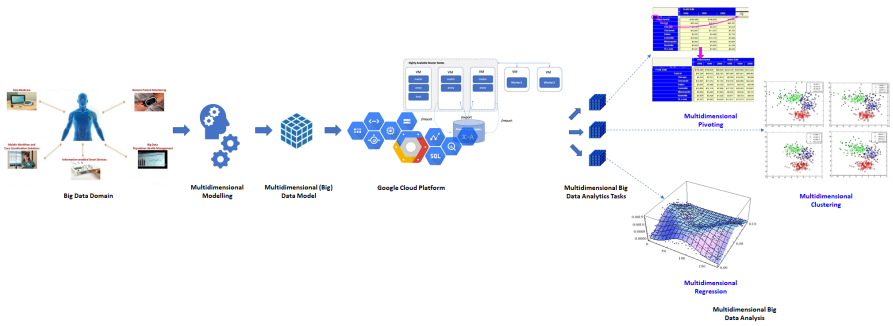


Fig. 3.1: Healthcare Data Cloud-based Processes

As shows the aforementioned figure, more recent data processes involving big data analytical tasks involve a data cloud layer leveraging cloud capabilities for the computation tasks required to generate analytics. First data are converted into models that can fit data specific databases (such as datawarehouses etc.) through the data modeling process. Next, and on top of the data model, cloud-based data processing is performed in order to generate analytics relevant to a certain type of tasks (supervised-based learning such as regression, or un-supervised ones such as clustering etc.). Next we define a few notions related to the topic of Privacy-Preserving Multidimensional Big Data Analytics so as to better comprehend the following subsections.

Published Works

[61] Alfredo Cuzzocrea and Selim Soufargi. "Privacy-Preserving Multidimensional Big Data Analytics Models, Methods and Techniques over Big Data: A Comprehensive Survey". In: *Expert Systems With Applications* (2023). Currently under revision.

3.1 Main Notions of Privacy-Preserving Big Data Analytics

The Figure 3.2, shows a hierarchical categorization of the main properties we next define. These properties are grouped according to whether they belong to the metrics or techniques sub-sub-categories of either data privacy or data analytics sub-categories. Such classification of the privacy-preserving big data analytics emphasises on the importance of the effectiveness of the described techniques by highlighting the related metrics by which each group of these techniques could be evaluated: For example, in order to assess the effectiveness of multi-dimensional OLAP-based data analytical techniques, query-based metrics could be leveraged.

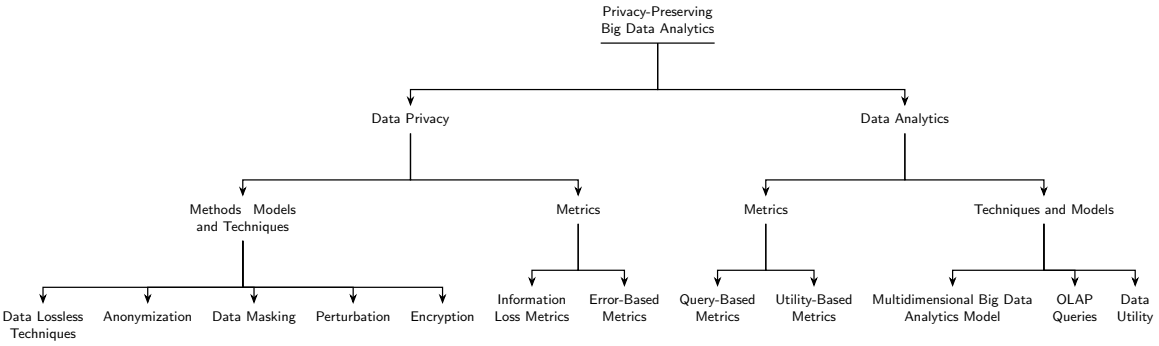


Fig. 3.2: Main Properties Groups

Data utility refers to the usefulness of the data for analysis, usually after anonymization. To capture the utility of the data, metric expressions may involve calculations over both the final anonymized data and the initial data. More specifically, a data utility measure should capture the difference between the distributions of data before and after the anonymization. Two notable aspects when assessing data quality through a utility metric are the information loss resulting from anonymization and the attribute importance. A well-defined utility metric should be able to capture intrinsic aspects impacting data quality for the considered application. For the sake of clarity, and since amalgamation is often made between data utility and data quality, we emphasize that the two notions are different. While data utility is useful to gauge before applying data mining techniques, data quality applies in a broader context. Indeed, data quality concerns features related to the data itself (as opposed to its usefulness for another process), such as data accuracy (for example, after a data sanitation process, it evaluates how different an attribute value is from its initial raw value), data completeness (evaluates the degree of missed data after sanitation), and data consistency (being the degree of attribute correlation holding after sanitation).

A data loss technique refers to a data processing technique that results in a partial data waste posterior to a "lossy" task (in particular, an anonymizing procedure). An example of such techniques is described in [197].

Data reconstruction refers to the set of algorithms that attempt to recover initial data that underwent a lossy process. As their name suggests, such processes incur a loss of data, so the main challenge of data reconstruction methods is to ensure that the entirety of the initial data is recovered (lossless reconstruction). Examples of data reconstruction methods are numerous in the literature, we mention [161]. Some researches are focused on a specific application of data reconstruction, as for example, for remote sensing images [216] or for recovering data in multidimensional time series [27], while others are more general and apply to any type of data as [109] where a method is based on a least-squares approximation in combination with a randomized Singular Value Decomposition (SVD) technique is proposed. In addition, certain other research works tackle data reconstruction in the field of privacy preservation: for instance, [86] describes a distribution recovery method for anonymized data that attempts to recover an optimal utility for users' sensitive data. Another worth noticing research work is [113],

Privacy-preservation needs stem from data confidentiality requirements that aim to protect the privacy of patients. These requirements are usually imposed by national and international data protection laws such as US Health Insurance Portability and Accountability Act (HIPAA) [38] and the (GDPR). Privacy-Preservation is established through a set of techniques and approaches aimed at transforming sensitive data into an unrecoverable format. Most often, privacy preservation is tied to the notion of data sharing. Indeed, data sharing is the practice of permitting a third-party to acquire the data to generate new insights and eventually enable fine-grained personalized

medicine. An important application in the healthcare area is anonymization, whose goal is to obfuscate the identity of individuals/patients to the greatest extent possible while keeping bonds with their sensitive information. To note that sensitive information should be minimally suppressed for data utility to be maintained. Examples of privacy-preserving methods include: k-anonymity [189] and l-diversity [134].

Data masking encompasses a set of techniques responsible for obscuring data in a matter of privacy-preservation. Contrary to other anonymizing techniques, data masking aims at concealing the data whilst keeping the initial structure of the attribute values intact using techniques such as Shuffling [99] changes the order of attribute values within a dataset column, Nulling (using generic symbol instead of the actual letters/numbers), Substitution uses a different synthetic (fake) attribute value to mask a target data value, Randomization [35] consisting of replacing attribute values with random data, and also Skewing, which acts on the attribute values through domain distribution variance to alter their initial authentic values. Blurring is also a data masking method that aims to alter each individual attribute value accordingly to yield updated values falling under a pre-defined range. A less common method is Averaging, where a summarizing aggregate relationship to the real data is maintained while certain attribute values undergo updates

Perturbation is a technique used to alter an attribute value for the purpose of anonymizing it. Typically, this technique leverages the use of a known distribution (e.g., the distribution of the data to anonymize) to alter the initial data, hence representing them in a privacy-protecting form (probability distribution-based perturbation). On the other hand, perturbation could be done in a "random" way using synthetically generated data to replace the initial values with new ones in a once for all manner (fixed data perturbation). In both types of perturbations, noisy data are added to the original data in such an ideal way to preserve certain statistical knowledge engraved in the data, such as the domain mean, variance, or correlation. Unfortunately, many literature perturbation techniques introduce bias (defined as the difference in query responses using perturbed and original data) when applied to the data, leading to the wiping of certain statistical information (summary measures) of the data, such as the mean, as well as the obliteration of correlations between the attribute values. Note that one perturbation technique that withstands all four types of biases is the General Additive Data Perturbation (GADP) [146].

Bucketization is an anonymization process that decouples equivalence classes (QID attribute values in record groups) from sensitive attributes yet keeps the link between the two for data analysis purposes. By loosening the bonds between the QID and the sensitive attributes, sensitive attributes within a bucket may be equally attributed to a given record (since they belong to a bucket and not to an ordered list anymore). Technically, the QID attributes are de-associated from the sensitive attributes, hence resulting in the creation of two tables connected through a group membership column

Group-ID. The Group-ID column is responsible for "roughly" keeping track of the origin of sensitive values and to which group of records they belong. A count of the sensitive values within each bucket is also maintained. Note that the count of different IDs in the Group-ID column represents the total number of generated buckets. This may lead to strengthening privacy and lowering data attack risks. An example of such methods is described in [37].

Recoding data means applying a privacy-preserving transformation over it (e.g., generalization). Locally recoding data is to permit one attribute value in a QID equivalence class (Quasi-ID) to have two or more transformed (e.g., generalizing) values. In other words, local recoding of a domain of values means the possibility of assigning different privacy-protecting (e.g., generalizing) values to the domain of values in a flexible way. The mapping of a domain of values to its generalized counterpart is performed on a tuple-basis. Indeed, attribute values are transformed (e.g., generalized) "by tuple" (and not by domain). More specifically, to locally recode a domain of values, a "neighboring" portion of occurrences will be considered for a certain generalization, while other "close" occurrence groups will be considered for other generalizing values. This flexibility in dealing with the domain values has the advantage of potentially securing more data utility by minimizing information loss during the anonymization process. Some notable examples of anonymization procedures leveraging local recoding are described in [11] and [208].

Global recoding domain values during an anonymization is to generalize all of occurrences of an attribute domain to a one generalizing value in order to reach the targeted anonymization threshold. Note that such methods may overgeneralize the domain values, leading to an increased data loss. The information loss resulting from the global recoding is much more important than of the local recoding. Additionally, global recoding could be considered a special case of local recoding.

Attribute-focused techniques aim to anonymize attribute values by performing attribute-related operations. Typically, this type of technique differentiates the anonymization procedure to apply depending on the type of the attribute (categorical or numerical). In both cases, the attribute-focused technique will enable the anonymization of the attribute values using two different approaches (one for numerical attribute values and another for categorical attribute values). Examples include interval width identification to group numerical values and the use of the mean for replacement of the target values. For categorical attributes, examples include an id-based matching technique [136] or clustering (of tuples through PAM k-medoid algorithm [156]) followed by tuple partitioning (where bucketed tuples satisfy an anonymization check, e.g., l-diversity) [153].

Contrary to attribute-focused techniques, ***data-range-focused*** techniques focus the attention on obtaining anonymity in target datasets via applying specific data processing procedures over ad-hoc collections of data items (stored in those datasets). The main idea here consists in devising models and techniques that intelligently exploit specific properties of data (e.g., dis-

tributions, ranges, maximum value, minimum value, and so forth) to achieve the desired anonymization. For instance, this is the case of aggregation-based anonymization (e.g., [30]), or obfuscation-based anonymization (e.g., [24]). Some approaches also combine multiple methods to obtain this effect (e.g., [167]).

Error-based metrics apply particular criteria that are defined starting from a given error metrics, which, in turn, can be native or derived from multiple (error) metrics. For instance, in the case of OLAP data cubes, a popular error-based metric consists of minimizing the query answer error (e.g., [58]), given a population of "typical" queries applied to the target data cube. Other initiatives make use of encryption techniques based on the maximization of decryption errors, in order to achieve the security-based privacy of target records (e.g., [7]), for instance in the healthcare setting. Furthermore, other approaches consider the error an hypothetical adversary could make in detecting dataset properties (like, for instance, the distributions of specific attributes in the target dataset), as in [102].

Query-based metrics are a type of metric based on answers by comparing those prior and posterior to the application of a transforming technique (e.g., anonymization, compression, etc.) on target data. This is important in order to inform users about the quality of transformed data and whether or not it fulfills a certain quality requirement. This metric may also be useful to quantify the data alteration caused by the applied transformation. For example, in the field of OLAP, query-based metrics are an assessment of the impact of a data transformation method (anonymization, compression, etc.) on the data utility/data loss (or information loss) balance. Further on this, OLAP query-based metrics could then evaluate such impact on the data by comparing the exact answer and the approximate answer (those submitted on the original data and the transformed data, respectively).

Information loss metrics are used to evaluate the degree of lost information during the anonymization process. More specifically, the metric evaluates the proportion of lost information due to anonymizing a certain domain. In the case of generalization, for example, it can be expressed as the max and min bound differences of a generalized domain over the same bound difference of the same domain. Whereas, in the case of suppression, the information loss metric can be expressed through the proportion of suppressed record count over the total record count for a domain. In addition, in the case of perturbation, information loss is measured through dissimilarity between the initial and the sanitized data. We mention, for the sake of an example of such metrics, dissimilarity [31], the generalized information loss metric [149], or also the discernibility metric [116].

Utility-based metrics measure the impact of anonymization on the utility of the data by considering the information loss for each attribute as well as their importance for data analysis. Differently from an information loss metric, a utility-based metric incorporates attribute weighting with respect to their utility for data analysis. For instance, we mention "Distortions of

Generalization of Tuples" [126], "Weighted Normalized Certainty Penalty" [207] or also "Total Information Loss" metric [34]. Other notable examples of data utility metrics are described in [118] and [117].

An aggregation method uses data collected from multiple sources in order to perform certain aggregation operations on them. In the field of privacy-preservation, aggregation may be performed over encrypted data in order to keep data safe while completing the aggregation task. The technique allowing such private aggregations is called homomorphic encryption. The aggregation is used to conceal the original data while enabling their later manipulation by the components of the anonymizing system. At the end of the process, aggregated data will be delivered to the end user in order to enable them to extract insights from the statistical properties that are within these aggregations. Method examples include the use of complex aggregations over encrypted data [131].

High-dimensional data are data that are characterized by an elevated number of features (or attributes), which is mostly the case in the healthcare domain. More broadly speaking, for a certain collected observation, the number of observed variables would be very high. For example, in the clinical data, if patient records are being the considered observations, then the number of record characterizing features (observed variables) is particularly high. This kind of data raises a lot of concerns about whether or not they could be effectively and efficiently handled (and most importantly anonymized), as this is remaining an open challenge nowadays commonly known as the curse of dimensionality. For example, it was suggested [10] that in order to privacy-protect high dimensional data it is necessary to suppress a high number of attributes, thus negatively impact the data utility and hampering a productive analysis of the data.

OLAP queries are requests submitted to OLAP data cubes that involve aggregations over designated target cells for the purpose of analyzing the underlying OLAP cubes. Most often, these queries are directed towards special servers known as Data Warehouses where the OLAP data cubes reside. We note important types of OLAP queries named range queries, where the aggregation operation is applied over targeted cells (containing the target measures) enclosed within the specified delimiting numerical ranges. For instance, the range-SUM family of queries computes the sum of the measures associated with cells within the specified boundaries. Another commonly used operation is the range COUNT family of queries, which counts the number of the specified range-enclosed cells.

A ***Centralized environment*** is characterized by algorithms or systems leveraging a centralized approach for storing and processing (e.g., data). All of the required operations are then done within the same location. Anonymizing algorithms leveraging such settings perform tasks on a single machine in a single location over an entire dataset (e.g., [189, 134]).

A distributed environment is where algorithms or systems leverage distributed capabilities for storage and computation. It usually requires the

use of a network to transport data or computed values to two or more geographically distant locations (e.g., client-server setting). In the case of data anonymization, distributed components may need to operate in parallel in an independent manner, where the anonymization task is shared across the participating "nodes" (often called "workers") in such a way that benefits the performance (e.g., processing time) while maintaining a good quality for the rendered anonymized dataset (e.g., in terms of utility). Many research works have dealt with distributed settings for anonymizing large datasets, for example, in healthcare [141] or more broadly for big data [73].

Scalability refers to a system's capability to cope with a larger workload by dynamically, adequately, and automatically self-allotting sufficiently needed resources (storage and computation) to function properly. A common heuristic that scalability-enabling algorithms may adopt, is to lower data loads to transfer over a network by considering a simplified representation of the considered data. A fact worth noting is that several literature works enable scalability through distributed architectures leveraging the MapReduce programming paradigm over the cloud, such as in [137]. Another example of scalable systems is described in [73], where a Mondrian-based anonymizing method was defined by extending the Mondrian algorithm [143] through an innovative quantile-based partitioning approach leveraging rankings of the dataset attribute values. In their approach, worker nodes are assigned exclusive portions of the dataset optimizing thereby data exchange rates between the nodes, which then can independently anonymize their respective local data fragments. The data portions are defined using a coordinator, which sets, accordingly, the conditions of the partitioning.

Multidimensional big data analytics model is a scheme enabling the multidimensional analysis of data by being aware of the underlying data structure. A considerable number of research works have defined multidimensional analytics models, such as in [182, 41]. A well-defined analytics model would be conceived based on the type of analysis sought and the application area. A well-known multidimensional data analytics-enabling tool is OLAP cubes. Indeed, an OLAP cube permits the multidimensional representation of data in a pre-analysis phase. Typically, OLAP is leveraged to build a suitable model for a target analysis application. For instance, [79] explicitly defines a multidimensional ROLAP data cube model and specifies star and snowflake schemas for air pollution data to analyze them and eventually enable decision-making and knowledge discovery. Another example is described in [173], where a terrorist data warehouse model is proposed. Indeed, a galaxy schema is defined to ease the investigation of terrorist acts for the specialists and to enable decision-making. The system also permits MDX querying over terrorist attack data to enable results depending on the investigator's intuitions. Furthermore, a privacy-preserving relating data analytics model is described in [69] which applies to the case of temporal open big data.

3.2 Big Data Seven Pillars of Privacy-Preserving Multidimensional Big Data Analytics

According to our view, the Big Data Seven Pillars of Privacy-Preserving Multidimensional Big Data Analytics are summarized in the following Figure. 3.3

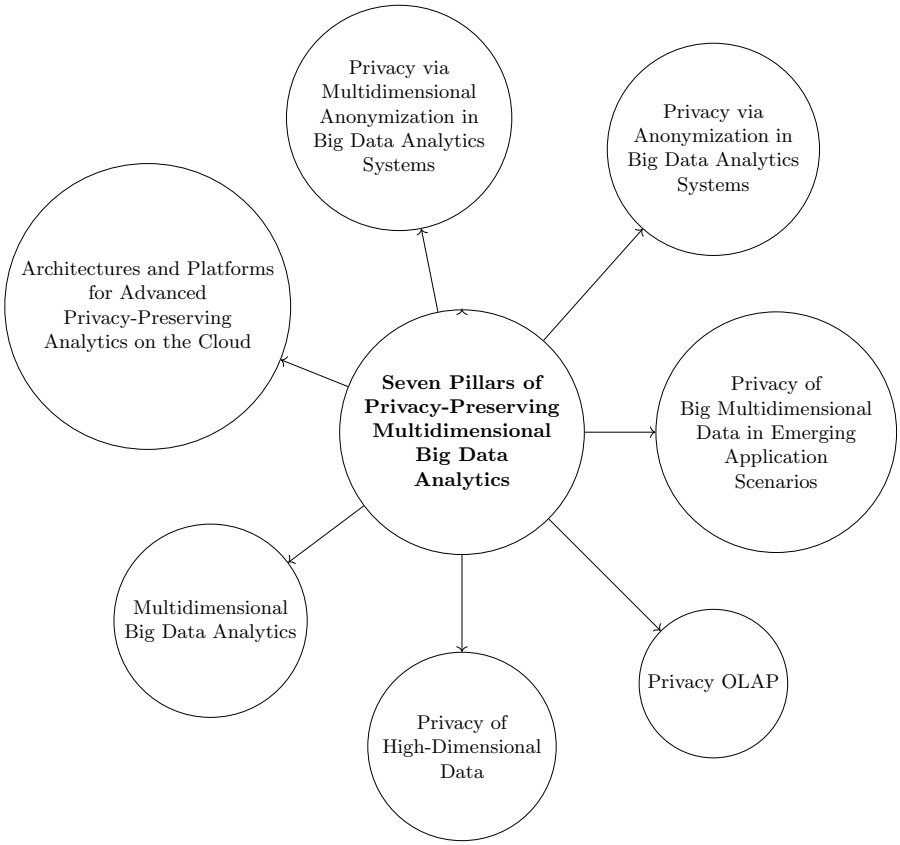


Fig. 3.3: 7 Pillars of Privacy-Preserving Big Data Analytics

In fact, privacy preservation big data analytics could be achieved in different ways and through different approaches and models. In the current work, we think of it as a topic that is composed of seven complementary sub-topics that draw the full picture of state-of-the-art privacy-preservation in big data analytics ecosystem.

3.2.1 Multidimensional Big Data Analytics

Multidimensional Big Data Analytics is the area where insights are derived out of data using enabling techniques and algorithms. Data from which the analytics are derived from, are usually stored in adequate storage systems such as datawarehousing technologies are relation databases based on suitable models such as OLAP or relational databases (tables or views) respectively. Big data analytics systems use a multitude of methods, techniques and algorithms (examples are decision-support systems [163] or also recommendation system [170]) to derive decision-making type of knowledge useful for a plethora of applications such as in the health, social media, tourism, entertainment among other domains. Examples of such systems are presented in [142, 174, 97].

3.2.2 Privacy of High-Dimensional Data

High-dimensional data are characterized by a count of variables that outnumbers the count of observations/samples. These kind of data are mostly considered by the literature as more difficult to secure [10, 9]. In response to this annoying challenge, some research works have dealt with it through a process called *vertical fragmentation* that breaks down large datasets into smaller portions of data having each fewer number of variables on which anonymization algorithms are more adequately applied. Indeed, vertical fragmentation considers attributes that demonstrate high level of correlation (or similiarity) to split them into seperate data entities on which state-of-the-art data anonymization techniques might be applied on [187, 22]. A basic example of vertical fragmentation is depicted in Figure 3.4. Examples of such works dealing with the *curse of dimensionality* problem are [125, 213, 206].

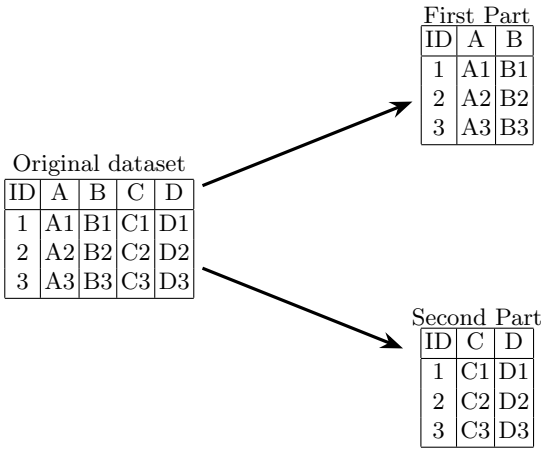


Fig. 3.4: Vertical Fragmentation Example

3.2.3 Privacy in OLAP

OLAP (acronym for Online analytical processing) enables multidimensional analysis of huge amounts of data. Specifically, this technology enables a multi-perspective and an interactive processing and analysis of multidimensional data and provides a fast analytics solution through adequately storing pre-calculated measures. Given the unprecedented surge in data generation as well as the need to analyze them via data warehousing technologies which use, potentially untrusted cloud services, privacy-preserving techniques for OLAP has become nearly unevitable. As an example, OLAP could be used to store and to analyze healthcare data in the goal of predicting diagnoses or to recommend therapies for patients. Literature works have dealt with a number of OLAP settings such as centralized data or even distributed OLAP [53],standard batch or even stream OLAP [67] these works were guaranteeing privacy in OLAP aggregations mostly through techniques based on either homomorphic encryption [48], sampling [56], data perturbation [144] or access control [90]. In short, OLAP privacy is crucial for a plethora of use case applications related to healthcare, finance, telecommunication domains to mention a few.

3.2.4 Privacy of Big Multidimensional Data in Emerging Application Scenarios

Emerging technologies [124] breakthroughs, especially in healthcare topics such as AI, blockchain or IoT, are shaping the way we use data. Regardless of the context and the environment in which data are leveraged to derive insights, privacy of data is constant in any system which should comply to by defining analytical processes that account for this parameter. As an example, in smart healthcare area, Electronic Health Records (EHR) are the new reference data source due to their added value to analytical outcomes and their considerable process improvement in the healthcare domain [202, 39]. An example [204] where EHR data are protected using blockchain-based access control technologies coupled with local a differential privacy technique. In such scenarios, we can deduce that the processes of privacy of big multi-dimensional data are heavily influenced by the 3Vs (Volume, Variety, Velocity) characterizing the data in such cases. In fact, in emerging scenarios where data analytics are generated, data are typically created in abundance at high velocity and are of various types and formats.

3.2.5 Privacy via Anonymization in Big Data Analytics Systems

Big data analytics are useful in decision-making [85] but prior to leveraging their analytical power, removing or altering identifiable information from the data at hand is mandatory. As this is becoming a requirement, making big data analytic systems more responsible for privacy entails being data privacy-aware. This means taking good care of hiding sensitive information when applying analytical procedures on the target datasets by incorporating a privacy

component. The privacy component will make sure the input data is processed into anonymized output data beforehand the analytics generation phase occurs. In this sense, [130] describes an anonymization method in the context of a BAN (Body Area Network) system, where wearable sensor devices are responsible for collecting electrocardiogram (ECG) data. The method is about adding noise to ECG dataset features-based on their order of importance to reach a well-defined differential privacy interference threshold. The interference threshold is updated on the fly as new data is measured from the sensory devices and takes into account feature importance. The interference threshold is responsible for guaranteeing a certain level of privacy for the data. Another example of such system is described in [215], where the proposed system utilizes Paillier Homomorphic Cryptosystem to prevent information leakage at the network channels level in a federated learning system context.

3.2.6 Privacy via Multidimensional Anonymization in Big Data Analytics Systems

Regardless of the setting or the assumption on the environment, data analytical systems are more constrained than ever to obey to data privacy rules. As seen previously, many algorithms have been designed to deal with privacy-preservation of data, but these are not necessarily tailored for highly-dimensional data as well. In the case of multi-dimensional data, many proposals have been made to tackle this problem: an example in smart grids field is described in [217] which employs a variant of the BGN encryption algorithm [120] to aggregate multidimensional encrypted data as well as an identity-based aggregate signature to guarantee. Furthermore, *Shamir* secret sharing technique is leveraged to help the transmission, in a fault-tolerant fashion, from smart meters to corresponding edge servers from where analytics are computed based on the Control Center submitted analysis requests.

3.2.7 Architectures and Platforms for Advanced Privacy-Preserving Analytics on the Cloud

Cloud services are more than ever exploited for a number of data-intensive processes useful to generate big data analytics. The extremely high volume of data generated through new technologies demands more resources to accomplish data-related tasks. In particular, privacy-preserving data analytics requires high computing and storage resources to perform well under data volume-constrained contexts. Examples are numerous, for instance in [159] a cloud-based video analytics as a service solution that protects video streams from side-channel attacks as well as from direct leakage of private video content is descrbed. Another example [209] presents a solution that uses differential privacy techniques to enable data sharing and analytics functionalities as cloud services. In Figure 3.5 we depict the most typical architectures for Advanced Privacy-Preserving Analytics leveraging the Cloud.

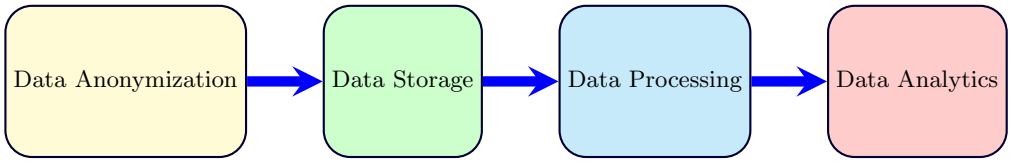


Fig. 3.5: Typical Data Workflow in Privacy-Preserving Big data Systems

In what follows a comprehensive enumeration of the most prominent literature proposal are detailed based on the Big Data Seven Pillars of Privacy-Preserving Multidimensional Big Data Analytics sub-topics.

3.3 Big Data Seven Pillars of Privacy-Preserving Multidimensional Big Data Analytics: Literature review

3.3.1 Multidimensional Big Data Analytics

Broadly, big data analytics could be created through mainly three data science ways: statistical summarization of the data, supervised learning based machine learning processes over the data or unsupervised machine learning processing as shown in Figure 3.6. In the following work [32], an unsupervised learning algorithm is proposed, specifically The paper proposes a framework for analyzing multidimensional data by applying clustering and classification algorithms to multidimensional views of input clinical data. The approach aims at finding anomalies in gestated human fetuses for early detection of diseases as well as for early detection of defective fetuses. The approach consists in building fetal growth curves, which would serve as benchmarks for comparing fetuses' growth and thereby spot anomalies. The main assumption consists in the fact that normal fetuses share similar baseline growth characteristics if they have the same gestational age, similar genetic properties, and have grown in similar environmental conditions. Baseline fetuses' growth curves facilitate the definition of groups of fetuses called Homogeneous Patient Groups (HPG). The membership of fetuses to one group of HPG will inform about their health (wellbeing and pathologies). Multidimensional views are created from target data, after that a classification algorithm is applied onto the data to detect the closest HPG group for a given fetal patient.

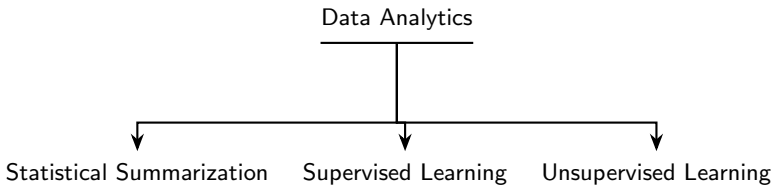


Fig. 3.6: Three Data Science Ways to obtain Big Data Analytics

Yet another relevant work based on OLAP analytics is described in [190] and proposes a solution to analyze multidimensional healthcare biological treatment-related data and to monitor their administration to the patients. They propose a micro-services-based tool where different components are deployed using container virtualization to analyze ROLAP-based data stored in the PostgreSQL database. They propose a dimensional fact model for analyzing biological drug administrations for the patients. This examples thus provides statistical summarization through OLAP operations on multidimensional data and enables a visualization components to better comprehend the resulting outcome.

3.3.2 Privacy of High-Dimensional Data

Privacy of high-dimensional data can be enforced in many different ways using simple or combined privacy techniques that leverage global or local recoding of the attribute values to conceal. Depending on the type of data, methods may vary in terms of the approach employed and/or the literature algorithms/algorithm variants used. In what follows, we describe few related works that relate to our realizations in this thesis. Firstly, in [94] a technique to capture the correlation of the data while anonymizing them, accounting for the maximization of data utility is defined and detailed. The example presented in this paper illustrates how the correlation matrix is generated, as follows: starting from "transactional data" made by some customers, the general matrix is built by placing a 1 when an item is within a transaction, a 0 if not. Deliberately, sensitive attributes are associated with sensitive items in order to ease the processing. They first start by converting the general matrix of the data into a band matrix through the Cuthill-McKee algorithm [50]. In addition, through their proposed greedy algorithm, CAHD, formations of sensitive transactions are created. CAHD works as follows: an iteration over a sensitive transaction consists of the grouping of p nearby (input privacy p degree) sensitive transactions together based on a notion of non-conflict attributes (two conflicting sensitive transactions have a common sensitive value) and through selecting together only transactions that have the closest number of common Quasi-ID. During the process of group formation, an histogram is maintained to guarantee that the constraint of privacy is satisfied. The algorithm halts

when no more ungrouped sensitive transactions remain, or when no additional groups can be formed.

Second, an utility-aware anonymization algorithm is detailed in [160]. Indeed, A multidimensional anonymization technique where generalization transformations are applied iteratively to the target dataset. The transformations are set in a lattice's vertices, and the algorithm traverses the lattice accounting for the maximization of data utility and a certain defined data risk threshold (a transformation is validated and used only if it increases data utility and if data risk thresholds are met). To achieve its goal, the algorithm toggles between a greedy traversal approach and a best-first search approach, indeed, if the lattice limit is reached, the algorithm performs a backtrack to be able to continue the traversal. If n steps were taken, the algorithm will also move to use the greedy algorithm, but here without performing any backtracking. While traversing the lattice, the algorithm maintains a queue where transformations are ordered according to data utility (descending order). Intuitively, the more the transformation preserves data utility, the less the associated data generalization.

3.3.3 Privacy In OLAP

OLAP is certainly one of the most adopted decision support and knowledge discovery solution in business intelligence [99]. OLAP is concerned with data privacy since it may incur the disclosure of potential sensitive information. As noticed earlier a number of literature proposals have defined frameworks and tecchniques to tackle OLAP privacy using privacy OLAP adated techniques like samping, perturbation, anonymization and so forth. In the following, we describe the most notable works in this topic. First, in [57], authors propose a three-step sampling method towards the anonymized analysis of data cubes, namely: storage space allocation for the samples extracted from the query workload (QWL) to obtain the approximative data cube answer, the actual sampling of the data query workload step and a refinement phase included to enhance the results in terms of accuracy (of the answer) and privacy (of data), thereby securing a balanced trade-off between the two target metrics. The authors also define techniques to deal with numerous problematic aspects hindering an accurate and private query answer for the data cube. For example, they propose to look at a query over a dataset as a set of mini-queries defined over a set of "regions", which are the bounding spaces of the mentioned queries. From there, they represent a query as a set of regions that should comply with several conditions to enable a fine-grained and effective analysis while keeping the queried data safe from ill-intentioned purposes. On the other hand, a data perturbation technique tailored to OLAP is described in [54] which proposes a novel technique to anonymize data cubes in a distributed environment. They use CUR matrix decomposition [81], a matrix decomposition method used to compute approximate representations of large matrices, to anonymize a 2D view of the distributed cubes. In the

distributed scenario, they propose a Secure Distributed OLAP aggregation protocol (SDO) [52] to anonymize the entirety of the data cube network. To stress the anonymization process achieved thanks to the CUR approximation, they perform an in-depth assessment of the reconstruction results by retracing the method first proposed by Agrawal et al. in Retention Replacement Perturbation algorithm [12]. The idea is to quantify accurately enough the values (through setting reasonable and rigorous bounds) of the yielded range-SUM query answers on the reconstructed data. A comparison between the mentioned query results over both a constructed version and the initial one is also performed to confirm the effectiveness and efficiency of the method used (CUR decomposition method). They also demonstrate some useful mathematical properties involving what they name as full-differential privacy (a quantification of the difference between the initial and the transformed view) and marginal differential privacy (a quantification of a partial difference between the initial and the transformed view on the queried attribute domain).

3.3.4 Privacy of Big Multidimensional Data in Emerging Application Scenarios

Nowadays, thanks to many underlying full-fledged technologies, many other technologies are in the state of fast development and are currently expanding. Those emerging technologies are technically to be further developed in the future given that their value potential is still unrealised. In the following we focus on few of such technologies as smart grids, eHealth or IoT while highlighting their added value for both data privacy and data analysis. Specifically, in [133] The authors propose a technique where the data are, first, packed through the Chinese remainder theorem and, second, encrypted through an encryption technique that doesn't rely on an established public-key encryption mechanism but on an approach based on key negotiation between users. The mentioned technique relies on time-stamping each user's data to achieve its goal. In addition, the encryption technique guarantees resistance against the infamous network replay attack. Phases of the scheme proposed, most importantly, include a data reporting phase (SM), an anonymized data aggregation phase (GW), and finally a data parsing phase (CC). Indeed, the developed system aims at collecting reports on electricity consumption from residential areas, including many residents (SM), to encrypt them and eventually forward them through gateways (GW) in an aggregated form. Their analysis is then done through Control Centers (CC). Given this setup, it goes without saying that the conveying of the data should be secure for the better of the users and the control center instance. The system is specifically tailored for a smart-grids environment and its main goal is to improve power delivery strategies for the residents. Secondly, a framework for optimal intelligent transportation services is provided in [158]. The main privacy component is ensured thanks to an innovative homomorphic encoding algorithm based on the Chinese remainder theorem and the Paillier encryption. The edge nodes

will receive ciphertext from the IoT devices, will perform homomorphic aggregations on them, and will send the aggregated result for analysis, at the control center. The Chinese remainder theorem was used to transform the input message vectors into integer vectors, and the Paillier method was used to actually encrypt these vectors into ciphertext. A more relevant framework to our project context, in healthcare area, is described in [219]. A double-securing privacy layer is defined: first a privacy-preservation technique is applied onto the initial data, residing in the healthcare center, second, privacy of data are further enforced through the analyzing query (using range SUM queries) which are submitted to the Cloud by the users/doctors. In order to facilitate the querying of the data through range SUM, an R-tree is employed to index the dataset. Authors claim that finding the intersecting data between a target portion of data (set in the traversed node) and the query intervals would serve as a good option for privacy-preserved range-SUM queyring. More into the algorithm details, the R-tree traversal is done in a depth-first way. Furthermore, the top-bottom search of the tree continues only if the query intersects with the current node's data, and eventually leaves are added to the query answer if they satisfy the mentioned condition. The authors assimilate the boundaries of each of the node's data and the query as integers and reason on their binary vector version to compare them, providing, thereby, an anonymized version of the evoked algorithm through the comparison of the data and query boundaries. Finally, the comparison would inform about the intersection of the query with the data (and thereby on the set of query answers). The comparison of the integers is transformed into an equality test based on a function they define, which varies according to vector variables set from the vector version of the integers in question. This comparison aims at concealing the intermediate operations (inequalities evaluation) results needed to obtain the resulting intersection between the query and the data. Specifically, they employ a homomorphic encoding-based data comparison technique (involving encryption of the vectors of data and the query and matrix-based calculations on the encoded versions of the vectors) that outputs the result of an equality test informing on the intersection between the original vectors.

3.3.5 Privacy via Anonymization in Big Data Analytics Systems

Data privacy techniques could be categorized as shown in Figure 3.7. As we have previously noted, data perturbation sums up in adding noise, independently, to attribute values in a way that yields values that are slightly different from the original ones. Examples include the differential privacy technique. Cryptographic approaches, on the other hand, use cryptographic algorithms to conceal attributes values. Examples include but are not limited to homomorphic encryption. Slicing approaches aim at improving the overall data utility while guaranteeing that the data are secured. Examples include the bucketization technique. Finally anonymization based approaches are used to protect user sensitive information by reasoning on the rows as to prevent disclosure

of such information through a plethora of privacy attacks such as record linkage, attribute linkage or also table linkage. Examples are k-anonymity or also l-diversity.

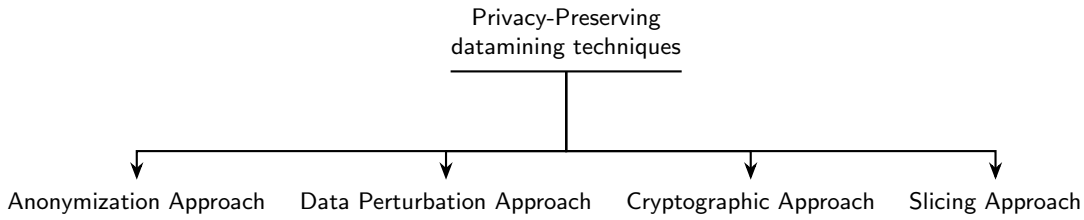


Fig. 3.7: Common Approaches to Privacy-Preserving Data Mining Techniques

In what follows, we review literature works that tackle anonymization realized in big data analytics systems.

In [141], first, a privacy-breaching constraint assumption is assumed which is that most data records piracy acts are enabled through the knowledge of the attribute values corresponding to only a portion of the entire QID set. Then, they propose a utility-aware privacy method named LKC-privacy. Namely, this method suggests that for each tuple of QID attributes of size lower than L included in the full set of QID, at least K records should share the same attribute values (guaranteeing K-anonymity). In addition, the probability of inferring a sensitive value within the equivalence classes should not exceed C% (guaranteeing C-de-association). They highlight two types of scenarios where anonymization of surgery and blood transfusion data are performed: a centralized approach, where a Central Government Health Agency (CGHA) anonymizes the data in LKC-privacy fashion for the recipient delivery, and a distributed approach, where the data are disjointly anonymized at each hospital performing the blood transfusion to the patients. It goes without saying that this technique enhances privacy guarantees over attribute values compared to basic anonymization techniques such as k-anonymity. As we are going to witness in the next examples, k-anonymity technique is considered to have weak guarantees over data safety thus a need to create a complex anonymization technique is the aim of most of literature works. Mainly, these works aim at combining algorithms and tools as to yield better privacy guarantees.

Next [152] a clustering based approach is proposed where data clusters consist in similar data points/tuples. Specifically, The authors propose to anonymize the data through a clustering-based approach using the α-de-association technique at data aggregating nodes on the network based on a client-server setup. Algorithmically, clusters are formed based on a distance metric evaluated between the tuples as well as based on the compliance of the

newly formed cluster (with the new considered tuple in the iteration) with α-de-association constraint. In addition to that, a server side anonymization phase ensures that clusters are merged according to their level of similarity to form a tree structure. The tree structure formed by nodes representing the clusters realises a higher level of privacy and ensures lower communication cost in a WAN, Internet, or Mobile network context.

Yet another approach based on clustering and using bucketization is described in [153]. The transformation for numerical attributes involves calculating, for each Quasi-Identifier (numerical attribute), the Interval Width (IW) (consisting of the mean of the largest interval of values of the considered attribute), after which the algorithm builds the equivalence class intervals that increment according to the IW from the smallest to largest attribute values. After building the ranges for the Equivalence Classes (EQs), initial tuple values are considered for mean calculation, leading to the anonymizing values that would replace the actual range of values of each EQ. In the case where attribute values, within one built EQ, are all equal, the algorithm proceeds to alter the value to conceal the original attribute value. In the case of categorical attributes, they propose an anonymizing bucketization approach based on a clustering sub-method that uses a k-medoid approach to create clusters out of the dataset to anonymize and partition the clusters (assimilated to buckets) to generate l-diverse sensitive buckets. The l-diversity sensitive attribute values check (sensitive buckets generation) is based on the calculation of the probability distribution of the sensitive values within each sliced bucket, which should satisfy an l-proportioned threshold.

3.3.6 Privacy Via Multidimensional Anonymization in Big Data Analytics Systems

Numerous works have dealt with the multi-dimensionality of data for anonymization involving much more complex processing. In what follows we describe in details few of them. [207], proposed greedy anonymizing algorithms that maximize data utility and minimize a proposed metric called weighted Normalized Certainty Penalty (NCP). Two approaches are proposed: the first one is the bottom-up approach, which starts by assigning each of the dataset tuples to a group and then tries to merge each of them (when the k-anonymity condition is not met) with another group and/or tuples (through a dataset scan pass) that would minimize the NCP metric. If the resulting merged group is larger than a certain threshold, then split it into smaller groups that satisfy the anonymization constraint related to the required group size (k). Distances were used for numerical attribute values and hierarchies for categorical attribute values to compute the NCP metric between the groups and/or groups of tuples. The second one consists of a top-down approach where the original dataset is partitioned into consequent subsets while accounting for the minimization of the NCP metric, indeed, the partitions should be able to be further partitioned into smaller groups of tuples that minimize the NCP metric. At

each iteration, partitions should verify the k-anonymity constraint. In the end, each group should satisfy k-anonymity, or an adjustment is performed.

This work [127] proposes an attribute hierarchical taxonomy approach for anonymizing the input dataset's attribute values through local recoding. The authors propose a new metric, namely Weighted Hierarchical Distance (WHD), to assess the generalization results of the initial dataset. The metric measures the ratio of the sum of weights of each of the domain levels between the initial value and the generalized value and the sum of the weights of all levels of the hierarchical domain. A distortion between two tuples (the initial one and the generalized one) is then deduced by summing over all of the WHD distances between hierarchical domain levels of each of the attribute values of the initial and the generalized tuples. The distortion between tables is then the sum of the distortions of each tuple of the table. Distances between two tuples as well as between equivalence classes are also defined. The former is calculated by, first, identifying the Closest Common Generalization (CCG) between the tuples, and second, considering the sum of the distortions of the first tuple with CCG and the sum of the second with CCG. Finally, the distance between equivalence classes is defined as the equivalence class size weighted sum of the previous distance (tuples within an equivalence class being identical). The paper then defines the main algorithm named: k-Anonymization by Clustering in Attribute hierarchies, which loops over all of the equivalence classes (clusters) of an input dataset, and iterates over those who have a size lower than k, spots the closest equivalence class (cluster) to the previously mentioned one, and generalizes (local recoding generalization) both of the equivalence classes (the currently iterated and its closest one). The algorithm halts when no equivalence class remains with a size lower than k. To help the local recoding anonymization by clustering, the authors introduce the notions of stub and trunk, along with a splitting technique to delimit the data portions to be generalized or processed.

Finally, in [218], The authors propose an iterative version of the Mondrian based on a MapReduce approach. A cached indexing tree, containing basic information about partitions of data, named Partition ID tree (PID-tree) is maintained and shared across the system nodes. To tackle excessive computational overhead that bigger dataset may incur, they propose to apply serial MapReduce to each new level of partitions the driver instructions may yield after a new iteration is completed. They propose an algorithm that accounts for two types of attributes: numerical and categorical. The driver "data splitting" iterations go on until all partitions satisfy a user-defined node's capability threshold. Over the course of an iteration, the algorithm seeks to yield a set of partitions optimally split for anonymization.

3.3.7 Architectures and Platforms for Advanced Privacy-Preserving Analytics on the Clouds

The solution proposed in [214] is a cloud-based e-health system enabling on-demand access to patients' EHRs (Electronic Health Records) while emphasizing the data privacy aspect. The proposed system architecture involves the use of third-party data centers to store data for later use as needed. To reach the end-users, data are conveyed over the network, where congestion and throughput issues are dealt with through data replication. The authors also propose a data acquisition model that allows fast and reliable data analysis that accounts for data processing time and priority. Besides, the data ingestion procedure accounts for the frequency of consultations of the patient and their health emergency. Indeed, a window-based temporary information for acquiring data is put forward in order to selectively delimit most informative data to secure. The architecture proposed comprises: an access control module that enables data access to users based on their roles and privileges, a cryptographic encryption module that is responsible for concealing EHR information into a cover image (through AES-128 algorithm and Steganography's LSB technique) stored in a third-party data center, as well as a steganography module that ensures that decrypted data are within reach of the data access module. Finally, upon request from a user, the access control module validates the data through the anonymization module by hiding sensitive values and anonymizing Quasi-ID records before the data can be disclosed and delivered to the requesting user.

In [155], The authors of this paper propose a system to gather data from IoT sensors for medical care purposes that is energy consumption "friendly". The data reach a medical server through a bridging network enabled by WI-FI, Internet, or cellular networks. In this paper, the integrity of data is secured via Homomorphic MAC (H-MAC) so that no malicious attacker can alter the content of the data transiting over the network. In this case, the receiving side (medical server) won't require the decryption of the sent message in order to perform computation on it. The authors believe that this will provide further protection over the data and will shrink risky network-related exposure of the data. The authentication process between IoT sensing devices and the coordinator (data generating side), as well as the latter with the medical server are secured using traditional hash functions and secret-public key mechanisms. In addition, the confidentiality of the data is guaranteed using homomorphic encryptions (known for enabling computations over the encrypted data). In fact, the system aims to reduce energy consumption by reducing the data transmission rate within the network, which accounts for 70% of the total energy consumption (the rest is mostly attributed to the data aggregation-induced energy consumption). To accomplish that, they use DPM [78], an approach that is set to train prediction models both at sensing nodes and on the medical center side to inform on which data are most relevant for sending to the medical server. The core idea they leverage is that if data could

be accurately enough (within the range of a defined threshold) predicted at the sensing node, then it could also be predicted at the medical server (since historical data used for training the models are the same for both sides). In addition, a simultaneous update of data-storing memories (needed for model learning) on both sides using predicted data values as well as newly sensed data values is executed. Provided this approach, it is possible to eventually differentiate useful from redundant data before sending them to the medical center. Eventually, pointless data transmissions will be avoided, thus engendering a reduction in network traffic and consequently a decrease in the total energy consumption of the system.

A slightly different approach based on clustering is proposed in [6], where outlier detection methodology is leveraged to prevent data from being disclosed. Specifically, The authors propose a cloud system enabling organizations such as hospitals or medical centers to accumulate anonymized data from patients. Their system summarizes in a cloud system provider that does in-house anonymization for later dissemination of the concealed data. The anonymization approach they adopt consists of the superposition of several anonymization-enabling steps that, they argue, form an effective and efficient clinical data anonymization technique. The core anonymization workflow sums up in the refinement and standardization of the data through the use of a normal distribution as a reference to spot outlier records and discards them as they constitute the most vulnerable records to the attacks. Second, they employ k-means++ [25] in order to cluster the records in a way that, they argue, is beneficial to data utility. Indeed, they explain that with such an approach, the anonymization of the data (through k-anonymity) is made easier and more effective as it requires less generalization as data entries are adequately clustered (based on QID values). Finally, they apply the k-anonymity method to each cluster separately, yielding thereby the anonymized chunks of data that union to form the final anonymized dataset ready to be delivered to healthcare organizations.

3.4 Summary

In the current chapter we have described our vision on how current privacy preserving for big data analytics approaches are implemented by providing on one hand a holistic view on their design and on the other on the key techniques they employ in order to attain their goal. We break down the privacy preserving big data analytics topic into sub-fields which we have detailed and provided literature examples as to show the relevance of our proposed structure of the current topic. We have specifically listed and detailed about literature works in the topics related to emerging technologies, cloud and broadly in analytical systems more importantly.

Starting from the following chapter we will introduce the set of techniques frameworks and solutions we designed and implemented for tackling the pre-

viously mentioned limitations of current state of the art and to enable better analytical processes in general that might enhance the healthcare processes and outcomes for what concerns cancer treatment and the quality of life after the treatment by providing tailored solutions that would improve the healthcare recommendation in terms of accuracy and effectiveness.

4

QFLS: A Complex Federated Big Data Analytics Learning System over Big Healthcare Data

The main goal of this chapter is to introduce, define and implement, a smart healthcare platform in the healthcare domain, that would typically ensure that the data being analyzed is anonymized and would enable a proper analytical dashboard for medical recommendation for the cancer patient in order to improve their QoL experience and general well-being after the treatments. In particular, and in order to reach our goal, we tackled hierarchical data analysis through a Tree Like Analytical Query, an innovative analytical querying approach based on attribute value constraints that provide statistical analysis, through aggregation, over the targeted data. Technical challenges that we have tackled through the current work include:

- Distributed aspect of data that are in different sources and the need for their analysis.
- Scalability of the solution (storage and computation capabilities) for large quantities of data
- Processing cost and resource optimization.

It goes without saying that the methodologies used heavily rely on the statistical analysis of data but in a novel way. In fact, the main analytical proposal is a tree-like analytical approach introducing the notion of lazy hierarchy (defined at querying time when the schema is detected (based on the schema-on-read paradigm)) approach which is free from strict containment of the attributes across the level of the hierarchy. Furthermore, and on the architectural level of our main solution, we rely on distributed storage system where data can be ingested at different locations preserving therefore their locality.

Nowadays, the excessively high volume of data that various technologies generate has become a critical cause of concern for centralized healthcare data analytics approaches (e.g., [165, 105]).

A trending solution that overcomes the previously mentioned issues [194], resulting from the adoption of a centralized approach for data analytics processes, is *data federation* [181], as another alternative to emerging *cloud computing solutions* [193, 112]. Data federation is the strategy of abstracting the

diversity of data sources into software that leverages data analytics. Systems adopting this concept will include a *software data virtualization layer* that abstracts the data sources into an integrated, easily accessible one. Data virtualization [192, 33] represents a data management approach that deals with disparate data storage in order to interface them into a unified, accessible, seamless centralized repository. The key task that data virtualization supports is the *extraction* of analytics from multiple data sources under a unified interface. In fact, data virtualization provides data with the flexibility to be federated. Contrary to data sharing, transfer, or pooling, the data federation paradigm facilitates data access and enables the effective on-the-fly querying of the data, with no-copy, in such a way that encourages collaborative statistical data analyses in a distributed environment. Data federation is also a solid concept on which federated data analytics [71] could be realized. *Federated data analytics* is a computing paradigm that enables performing analytical related computing tasks across multiple decentralized locations where the data permanently reside. This approach aims at supporting statistical and analytical needs such as deriving aggregated results through running local computations on the target data. A conceptual federated analytics model is depicted in Figure 4.1 where a cloud based system queries disparate data sources to gather relative analytics from each of the target federated data storages. Once at hand, local analytics are aggregated and transformed into global analytics that may inform and support decision making and to eventually enable recommendation over treatment administration and patient monitoring. This, as a consequence, will improve the overall patient care and avoid adverse events resulting from incorrect, in dosage or timing, of treatments injected to the cancer patients.

Federated learning has spawn as an approach to permit the processing of multiple distributed datasets while allowing full-access to the data only to the data owner. In fact, in a federated setting, data cannot be tampered with from any party except the data owner. The data processing is achieved in each of the distributed localities and the central server would gather partial analysis to process and yield a global one. The main advantages of a federated approach to data processing are:

- Improved scalability and cost-efficiency (by enabling the extension of data nodes and by preserving data locality; e.g., nulled data transfer costs).
- Improved overall data privacy (mainly by preserving data locality; e.g., reduced network based attacks).
- Improved data analytics diversity and accuracy (by learning from many data domains/organizations; e.g., widened range of data domains).

More broadly speaking and in addition to the privacy issues that challenge the classical centralized approaches [40, 93], these also impede data FAIRness principles [203, 194].

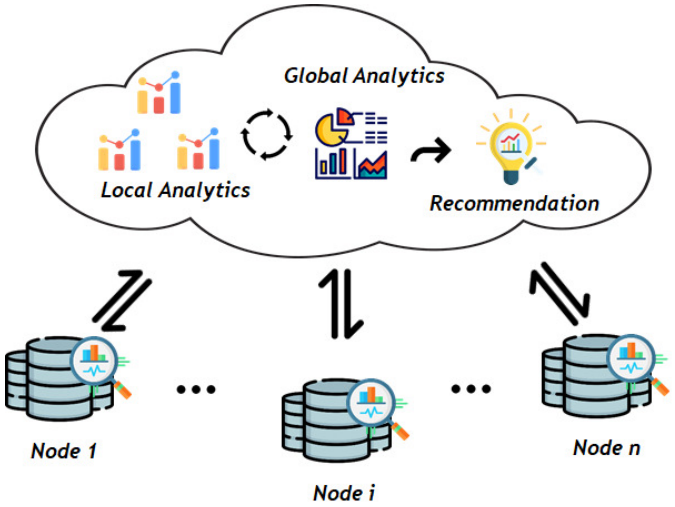


Fig. 4.1: QFLS Federated Processing

In the current research, we have conceived a federated system enabling federated analytics extendable to a federated learning system (by incorporating machine learning processing at each node of the federated system).

Key Contributions

In this chapter, our main contributions consist of the establishment of a robust healthcare data analytics system by presenting *QFLS: A Federated Learning System over Big Healthcare Data*, which encompasses architectural design, efficient data modeling, practical implementation, governance policies, machine learning integration, and adaptability to heterogeneous data sources, thus paving the way for more effective and comprehensive healthcare data analysis.

Overall, in this chapter, we make the following contributions:

- We initiate by establishing the foundational architecture for the QUALITOP data lake, which serves as the backbone for the subsequent research endeavors;
- We define and rigorously test the data model tailored for the analysis of distributed hierarchical healthcare data – this model is designed with a focus on efficiency and effectiveness;
- We address the critical functionality of adapting the system to accommodate anonymous data sources – this adaptability is crucial in real-life scenarios where data sources cannot be disclosed due to *privacy-preservation constraints*;

- We test our proposed QFLS system over a real-life case study, where data are gathered from different and heterogeneous data sources;
- We outline plans for enhancing the efficiency of the data lake. This includes the incorporation of a data indexing component, which can significantly optimize data retrieval, and the establishment of a data governance policy.

Chapter Organization

The remaining part of this chapter is organized as follows. Section 4.1 illustrates the QFLS context, architecture, and implementation details. In Section 4.2, we highlight the setup of the QFLS core cloud-based system. Section 4.3 provides a comprehensive description of the developed the QFLS anonymized dataset population tool. After that, in Section 4.4, we present the QFLS anonymized dataset analytics tool. Section 4.5 focuses on an innovative case study where we describe and present how the QFLS system operates and functions over a real-life scenario. Finally, Section 4.6 contains summary and future directions for this investigated context.

Published Works

[62] Alfredo Cuzzocrea and Selim Soufargi. "QFLS: A Cloud-Based Framework for Supporting Big Healthcare Data Management and Analytics from Big Data Lakes: Definitions, Requirements, Models and Techniques". In: *Proceedings of the 12th International Conference on Data Science, Technology and Applications, DATA 2023, Rome, Italy, July 11-13,2023*. SCITEPRESS, 2023, pp. 422–428.

[63] Alfredo Cuzzocrea and Selim Soufargi. "QFLS: A Complex Federated Big Data Analytics Learning System over Big Healthcare Data". In: *Big Data Research Journal* (2024). Currently under revision.

[64] Alfredo Cuzzocrea and Selim Soufargi. "Supporting Big Healthcare Data Management and Analytics: The Cloud-Based QFLS Framework". In: *Big Data Analytics and Knowledge Discovery - 25th International Conference,DaWaK 2023, Penang, Malaysia, August 28-30, 2023, Proceedings*. Vol. 14148. Lecture Notes in Computer Science. Springer, 2023, pp. 372–379.

4.1 QFLS: Context, Architecture and Implementation

In the ever-evolving landscape of healthcare, the efficient and secure utilization of vast amounts of healthcare data has assumed paramount significance. The emergence of big data and the need for privacy-preserving methodologies have given birth to innovative solutions like Federated Learning. In this Section, we delve into a detailed examination of *QFLS: A Federated Learning*

System over Big Healthcare Data. This comprehensive exploration is directed towards providing a deep understanding of the context, the architecture, and the practical implementation of this innovative system.

Figure 4.2 shows the main blueprint of the big data processing flow supported by QFLS. Here, we introduce a two-node-based scenario: one QFLS federated node, located in France, and one QFLS core node, located in Italy. Therefore, the French node provides (anonymized) healthcare data, and the italian node supports big data analytics and predictive analytics over these data.

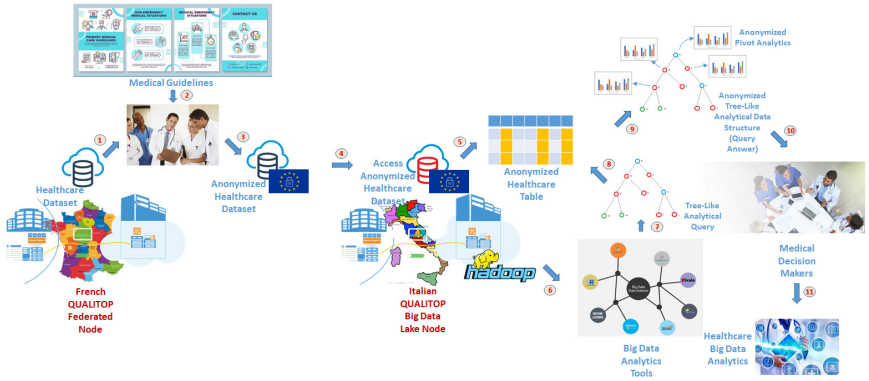


Fig. 4.2: QFLS Main Blueprint

As shown in Figure 4.2, at the French node (1), where the target healthcare dataset D is located, medical operators generate an anonymized version of D, denoted by D' (3), according to specific medical guidelines (2), yet compliant with the *GDPR*. Then, the analytics tools located at the italian node (4) execute aggregate queries, defined by the input TLAQ (8), which, in turn, is driven by target big data analytics tools defined within the QUALITOP big data lake over the remote french node via federated query algorithms based on Apache Spark, and the anonymized representation of the result is so-obtained (5). The final TLAQ analytics answer is shaped as a tree (9) such that each node indexes a proper anonymized healthcare dataset, empowered by the cloud computing potentialities (e.g., distribution, indexing, load balancing, mirroring, and so forth). The latter data structure is accessed by medical decision-makers (10), who provide the final big data analytics and predictive analytics (11).

In the current work, we focus on batch data processing (vs stream data processing through the *Speed Layer*) as to generate analytics from the ingested data based on a lambda-like architecture. A comprehensive illustration of such architecture is shown in Figure 4.3. Mainly, this architecture considers a datalake as the superposition of layers that will provide the right tools

and functionnalities as to ingest, store, process and serve data for analytical purposes to the users.

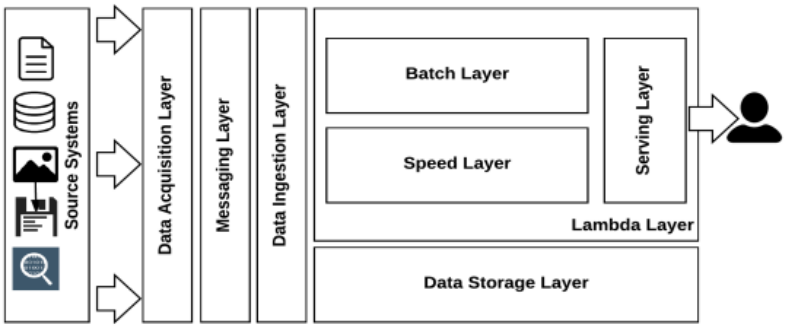


Fig. 4.3: Lambda Architecture [115]

Indeed, stream data analysis are out of scope of this research work, as we mainly deal with batch processing of data ingested through the corresponding layer.

4.1.1 QFLS Architecture and Implementation

Through this current work, our end goal is to design a data lake that is aware of the distributed aspects of the data sources. To the best of our knowledge, considering that there is no literary consensus on what data lake architecture provides. Instead, data lakes could be designed according to the specific needs of the organization. Nevertheless, data lakes should comply with specific standards. Indeed, a data lake must include common and well-known features capable of achieving the primordial goals for which data lakes were invented in the first place, such as:

- Easily scalable;
- Secure;
- Data analytics enabling;
- Allowing data ingestion;
- Utilizing a schema-based read approach;
- Offering the accessibility and findability of data through a proper data governance mechanism.

Some literature research works have addressed the issue of defining a data lake architecture. For example, the lambda architecture [115] enables the differentiation of processing of batch and stream data, where the tools and approaches used for each type of data are different. Another example of an

architecture that is based on the type of data to process is described in [164], where these *zone* architectures define the set of tools and approaches to be applied to the data depending on the degree of their actual processing. Yet another type of common architecture is based on *ponds*, which consist of an idea similar to *zone* architecture except that the latter differs in the data flow approach (data are localized at one pond at a time). An example of such architecture is described in [107]. More generally, some research has attempted to define data lake architecture frameworks (e.g., [95]).

In the *QUALITOP* project, a skeleton for a conceptual data lake architecture tailored to the healthcare domain has also been defined. In the current work, we extend upon it to implement the basic components of what would assemble into an efficient and effective data lake. In the following, we highlight what would be the first building blocks of the QUALITOP data lake, which represent the components of our QFLS system. Those will guarantee, most importantly, scalability, security, and data accessibility for the purpose of enabling data analytics. The open-source software *Apache Hadoop* (v. 3.2) [2] is the core data processing enabler of the QFLS system. As previously hinted, QFLS is a distributed system that allows the ingestion and remote querying of the datasets at each of the federated nodes. In the following, we detail the proposed three-tier cloud architecture.

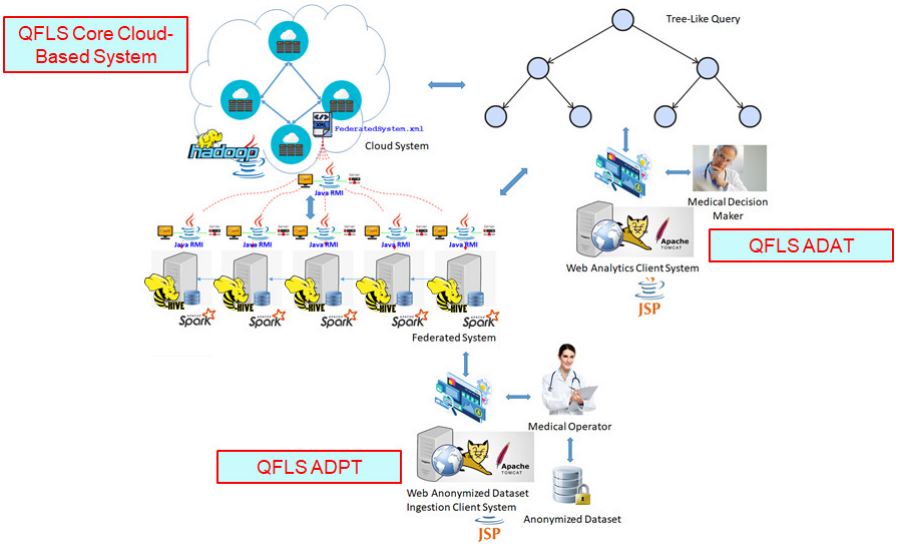


Fig. 4.4: QFLS Architecture

As shown in Figure 4.4, our system is mainly composed of three subsystems and components that interact in a distributed environment. In the next, we will detail each component.

- *QFLS Core Cloud-Based System.* This core system is, basically, a Hadoop-based cluster that serves primarily as an engine for processing Map-Reduce logic implemented at the federated node servers and is also housed for the configuration of the system. The Map-Reduce requests for data processing are received at the RMI servers, running in each of the federated nodes, after user interaction through the analytical enabling web interface;
- *QFLS Anonymized Dataset Population Tool (ADPT).* The ADPT tool permits data ingestion within QFLS. Indeed, QADPT is deployed at each of the federated nodes and uses a Hive database to persist data at each node while providing an intuitive interface to list, upload, and delete datasets. This web-based application is only used by medical operators in order to inform about the data on the actual federated node. To note that, since Hive uses Hadoop, the data are stored (i.e., copied) temporarily in HDFS for the time of processing and then deleted for privacy matters;
- *QFLS Anonymized Dataset Analytics Tool (QADAT).* The QADAT tool is the main tool enabling the analytics. It consists of a web-based application interacting with a Hadoop cluster in order to enable data scientists and medical staff to explore and analyze data ingested within QFLS. The tool provides an access control mechanism that accepts or denies access to QFLS components and also to the data, depending on user roles. As well as including a local database useful to save past query executions.

Focusing more on implementation details, the Hadoop-based cloud system hosts the QFLS system configuration that informs the client application about the dataset content of the federated system. Also, it represents the main processing engine used to execute Map-Reduce jobs submitted from federated nodes through the RMI server. As we have already pointed out, within each federated node, anonymized datasets are ingested through a data population tool. In addition, at each federated node, a running server receives the requests from QADAT and runs Spark-based Map-Reduce processing on the cloud. Indeed, the Hadoop cluster is responsible for processing all of the simultaneous Map-Reduce requests that could be received by each of the queried federated nodes. Note that the federated system nodes are the data entry point to QFLS through QADPT, and therefore they define the storage layer of QFLS. The ingestion layer sums up the available set of QADPT tools in order to populate the QFLS system with anonymized datasets.

More broadly, federated systems are known to facilitate the collaborative analytical processing of data, which in turn unlocks integrated data insights. For instance, medical decision-makers could take advantage of this architecture to identify common factors causing particular symptoms of a disease across several data sources. From a privacy point of view, this architecture has the advantage of preserving data locality while enabling the clinical insights required for specialized and tailored patient treatments. Preserving data locality is intrinsically linked to increased guarantees for the safety of the data.

Furthermore, as a result of this architecture, each registered node may adopt a different anonymization procedure to protect the data.

One open issue consists of the costly remote access to the federated node datasets. Indeed, data transfers, however optimized they could be, may constitute a critical issue in a geographically distributed federated learning system like a *WAN* eco-system, where locations are widely distributed across countries (or even continents). In our case, remote access to the distant federated nodes is enabled through the *RMI* Java technology [80], which serializes the Java objects that could then be sent over the WAN network. However, due to data access restrictions, the *data discoverer module* (DFD) performs all of the data processing on the data server-side and then receives the resulting analytics after executing the associated tasks within the federated node (using Hadoop). For what concerns the enabling tools and technologies, the transfer of the data to DFD is ensured by additional software, namely the *RMIIO* (v. 2.2) [154] package, which is an *RMI* utility that supports the streaming of a large volume of data on top of the *RMI* framework. Furthermore, as highlighted previously and as shown in Figure 4.4, the DFD module is guided through the *FederatedSystem.xml* file located in *HDFS*. *FederatedSystem.xml* is an XML-based file that describes the location, type, and name of each of the datasets ingested into QFLS. More specifically, the network addresses (IP) as well as their datatypes (CSV, TSV, and PSV) and their names are defined by the mentioned configuration file.

The server running on each federated node is implemented using the *Apache Spark* (v. 3.1) and is configured to execute over Hadoop cluster using *Apache YARN resource manager* [211]. Proper configuration for the Hadoop cluster resource sharing was adopted to enable concurrent Map-Reduced task execution received from multiple federated nodes at a time. These configurations include the right balancing of executor nodes for CPU and memory across the federated nodes. Finally, each of the developed client applications is deployed using an instance of *Apache Tomcat* (v. 8.5) at both ends of the QFLS system. To note that since each of the federated nodes runs a Spark-based server instance, it is possible to control the data type to be used for reading the data before the data processing phase occurs. Indeed, the data type is automatically detected, based on the dataset name that belongs to the list of present datasets in the node, and Spark proceeds with reading, in the adequate format, the target data to be analyzed (e.g., whether the target dataset in the format *JSON* or *CSV* or others formats supported by Spark). This setting would enforce data heterogeneity processing capabilities for *QFLS*.

4.2 QFLS Core Cloud-Based System

In this Section, we highlight and describe the setup of the QFLS core cloud-based system, which is a set of virtual machines configured as a Hadoop clus-

ter (one *NameNode* and multiple *DataNodes*) according to the Map-Reduce paradigm. The main idea is that portions of data are processed concurrently at each of the participating containers (running JVMs) through data shuffling and ordering operations. The partial data processing are then performed on the slave machines. After that, data portions are reduced in order to form the final transformed data that are eventually sent to the master node. All the data transfers (between the data nodes and the master node) are secured through data reading/writing operations. As noted previously, this component is responsible for executing submitted Spark jobs by each of the federated node servers in a Map-Reduce fashion. Figure 4.5 shows the Spark jobs executed based on RMI call requests issued by the QADAT client.

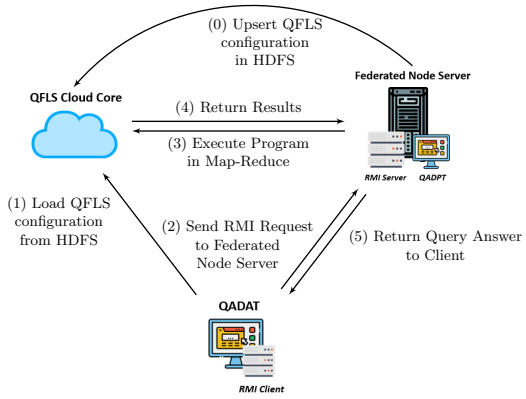


Fig. 4.5: QFLS Map-Reduce Program Execution Workflow

A description of each QFLS Map-Reduce program execution step is provided in what follows:

1. The QADAT client tool will read QFLS configuration from the cloud (HDFS);
2. After the system becomes aware of the locations of the datasets, it can offer the user an available selection of datasets to query through RMI requests;
3. The RMI server dispatches the processing to the cloud (for a Map-Reduce-based processing);
4. The cloud returns the computed results to the federated node RMI server;
5. TLAQ answer is then returned to the QADAT client.

4.3 The QFLS Anonymized Dataset Population Tool (QADPT)

Healthcare data are typically fragmented for a myriad of reasons, including the complex structure of the healthcare system and processes. As a result, healthcare data are becoming more and more available and are being created from various sources, such as clinical institutions, patient-generated health data, research, and public health. Furthermore, and more specifically related to our QUALITOP project use case, data sources vary in terms of location (e.g., QUALITOP project partners may each generate their own data for analysis). In this context, there is a need to manage a distributed storage system from which huge volumes of data are ingested. In order to address all of these converging drawbacks, QFLS allows users to ingest data at each federated node through the dedicated QADPT tool.

In this sense, the *QFLS Anonymized Dataset Population Tool* (QADPT) is a web-based client application designed to enable the ingestion of anonymized datasets into QFLS federated nodes. At each node, QADPT is deployed to facilitate the ingestion of datasets and their storage in an adequate, large database. The QADPT employs a user-friendly web interface to enable the ingestion, browsing, and deletion of datasets within each federated node. As a result, QADPT is a tool dedicated to populating each of the federated system nodes in a safe and simplified manner. Indeed, QADPT plays a technical medium role between QFLS and the data generation entities and organizations. QADPT assumes that the datasets being incorporated into QFLS are already anonymized; therefore, no additional anonymization or privacy tasks of the data are required to be performed prior to their actual ingestion into QFLS and their storage in a federated system node.

4.3.1 QADPT at Work

In this Section, we delve into the working methodology of the QADPT tool, meticulously outlining its operational capabilities. We further elucidate its practical user interactions through a *comprehensive use case* and a detailed *sequence diagram*.

The anonymized dataset types span a wide range of file extensions, but our focus consists of CSV, TSV, and PSV types since anonymizing such datasets is easier using our tool. Moreover, adding or deleting a dataset will trigger the automatic update of the system configuration file (in HDFS). Another characteristic of our tool is that it provides users with the current federated node content statistics (i.e., the number of datasets included in the node, their total size, the total count of records, and the average size of a dataset contained inside the node) in each of the QADPT interfaces. As well as sending a notification message after each successfully completed process. On the other hand, whenever the medical operator uploads a dataset, a Hive table is created, and the data is loaded into that table for eventual usage by the federated node

server. Additionally, Hive table creation and data loading are supported via a JDBC Java client. The implementation of QADPT was achieved using the Spring framework v. 5 (Java v. 17), *Apache Hive* (v. 3.1) [36], and *JavaScript language* (JQuery v. 3.6) [169].

Listing of Datasets

One of the main functionalities of the QADPT is that it facilitates the comprehensive display of anonymized datasets stored within the node. This feature enables medical operators to readily access and review the available datasets, along with pertinent information such as dataset name, size, and upload time. Additionally, search functionality enables operators to efficiently locate specific datasets by name. Furthermore, QADPT provides valuable statistics regarding the node data content, including the total number of datasets, their cumulative size, and the overall count of records. Figure 4.6 illustrates the QADPT interface for listing and displaying datasets.

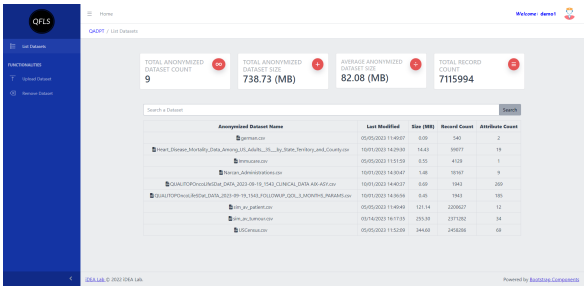


Fig. 4.6: List of All Datasets

Deletion of Datasets

Complementing the display functionality, QADPT offers an efficient mechanism for deleting stored datasets. Similar to the display interface, the deletion interface also provides comprehensive statistics regarding the node data content. To permanently remove a dataset, the medical operator simply has to select the target dataset and click on the corresponding delete button. Upon the deletion phase being completed, the *.ini* QFLS configuration file, residing within HDFS, is updated to reflect the revised dataset collection. Simultaneously, the node storage system (Hive tables) is also updated in order to maintain data consistency. The deletion QADPT interface is presented in Figure 4.7.

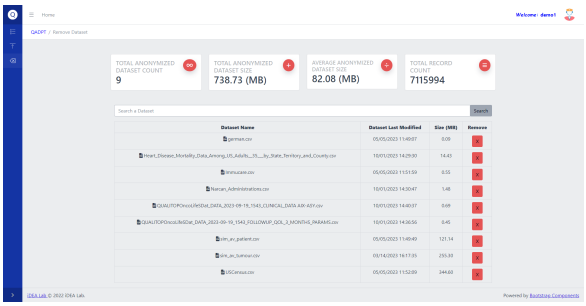


Fig. 4.7: Delete a Dataset

Uploading Datasets

Another main functionality of QADPT consists of the uploading process of anonymized datasets to the current node. Medical operators can select *CSV*, *TSV*, *PSV*, or other supported data formats for uploading. One of the advantages of our tool is that it provides an intuitive and simple interface to use by offering two types of functionality: a file chooser and a drag-and-drop component. In addition, a progress bar displaying the upload progress is available within the same interface. Upon successful upload, the QFLS configuration file is updated to reflect the newly added dataset, and the ingested dataset is then inserted into the corresponding Hive table. Figure 4.8 shows the QADPT interface responsible for uploading datasets.

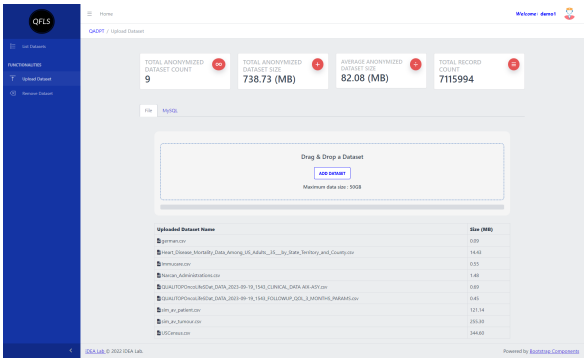


Fig. 4.8: Upload a Dataset

QADPT Use Case Diagram

The QADPT use case diagram provides a high-level overview of the system functionalities and the interactions between users and the system. Figure 4.9 depicts the use case diagram of QADPT.

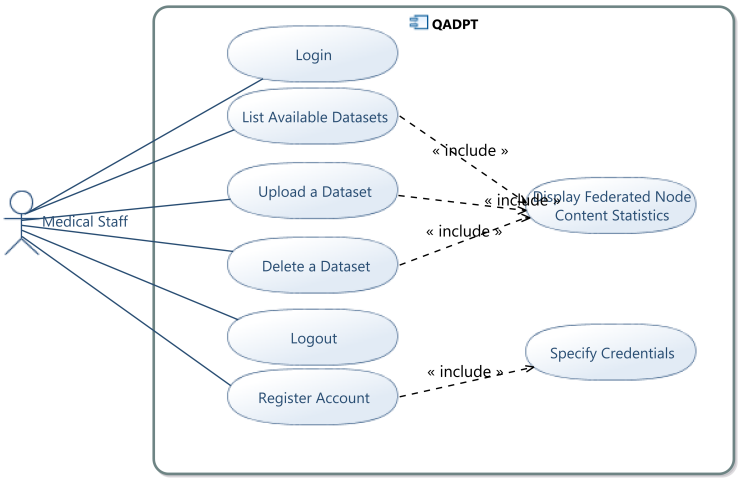


Fig. 4.9: QADPT Use Case Diagram

As shown in Figure 4.9, it is worth noting that only medical staff could create an account in order to login and load data into the federated node. Also note that all listing, deleting, and uploading dataset interfaces integrate a node content summary, which appears on the upper side of the interfaces.

QADPT Sequence Diagram

The QADPT sequence diagram serves to illustrate the detailed interactions between various system components and actors over time, helping to understand the behavior of the system. Particularly, the diagram highlights the interaction between users, QADPT, and both Hive and Hadoop HDFS. The corresponding sequence diagram for QADPT is shown in Figure 4.10.

As presented in Figure 4.10, specifically, we put emphasis on the fact that we have used Hive tables to store data (eventually storing them into HDFS) and that at each performed operation of uploading or deletion, the system updates its configuration.

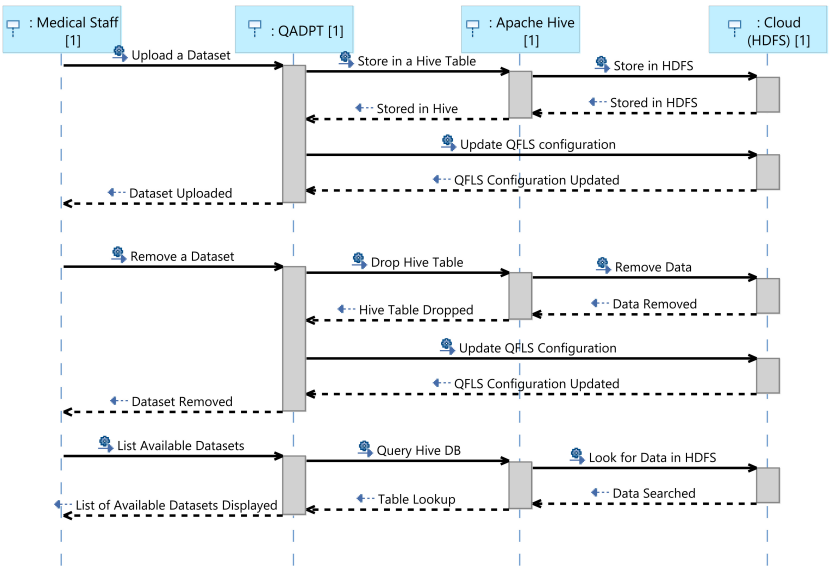


Fig. 4.10: QADPT Sequence Diagram

4.4 The QFLS Anonymized Dataset Analytics Tool (QADAT)

In this Section, we present and well-describe the *QFLS anonymized dataset analytics tool* (QADAT), which is one of the main components of our QFLS federated learning big data analytics system responsible for performing big healthcare data analytics.

For the purpose of enabling the analysis of distributed healthcare data in the QUALITOP data lake context, we designed a web-based client application to support big healthcare data analytics by providing statistical information on geographically dispersed data in a federated analytics system context. This web-based client application enables the analysis of a distributed set of anonymized datasets. At each federated node, different types of data are queried, generated from various sources, and ingested at different federated node servers.

QADAT employs an intuitive web interface to enable the analysis of a distributed set of anonymized datasets. The application has a *role-based access mechanism* consisting of providing different interface modules for each of the considered roles (i.e., medical staff and data analysts). These two roles have in common the following four main modules:

- The *Data Federation Discoverer Module*;
- The *Metadata Analysis Module*;

- The *Data Analysis Module*;
- The *Predictive Analytical Environment Module*.

Moreover, another dedicated interface for the exploration of past query answers is available to the users.

In what follows, we assume that healthcare data are mostly hierarchical in structure; thus, we adopt an analytical model able to analyze these data in an intrinsic manner.

4.4.1 TLAQ

In this Section, we introduce the well-known *TLAQ analytical model* used to perform the hierarchical investigation of healthcare data. As well as presenting its execution methodology and providing an example of TLAQ analytics over the *IMMUCARE* dataset.

Definition

TLAQ is an innovative analytical model designed to hierarchically investigate target datasets with the purpose of analyzing them. TLAQ constraints are based on potentially unrelated, not strictly contained, attributes. Equation (4.1) provides a definition of a tree-like query.

$$Q = \langle N, DS, A, AP, \langle Cond \rangle \rangle ; Cond = \emptyset | \{ \langle B, SP \rangle \}^* \tag{4.1}$$

where: N represents the name of the federated node, DS is the dataset located in N, A is the attribute in DS on which the predicate AP is applied, AP is the aggregation predicate applied to A, B represents the attribute in DS on which the predicate SP is applied, and finally SP is the aggregation predicate applied to B.

Analytical Query Execution Methodology

In this Section we delve into an exploration of the TLAQ execution methodology. The idea consists of creating a *tree of constraint-based nodes*, where each node specifies an analytics-related constraint converted into a *SQL SELECT* operation followed by a predicate (e.g., *COUNT*, *AVG*, etc.) to derive a statistical result from the targeted data.

On the other hand, the results are displayed in form of a *SQL view*, where only specified visualization attributes are considered in the final output data. Since the SQL query can only be executed with all required attributes, removing attributes that are not included in the visualization set would make the subsequent node constraints invalid. In order to mitigate and address this issue, we perform a *JOIN* operation between the current node dataset and the initial dataset before performing subsequent node processing.

Figure 4.11 provides a visual illustration of the analytical query execution methodology for *TLAQ*, and the related pseudo-code algorithm is presented in Algorithm 1.

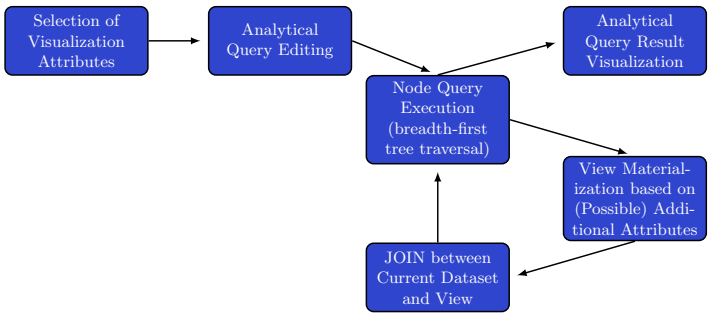


Fig. 4.11: Analytical Query Execution Methodology

Algorithm 1 TLAQ Execution Methodology

Input: *Dataset D, Constraints C*
Output: *AnalyticalResults AR*

```
AR ← null;
DV ← null;
VA ← null;
T ← null;
VA ← selectVisualAttributes(D.attributes());
T ← createTLAQAnalyticalQuery(C);
DV ← D;
for node in T do                                   ▷ Breadth–first search (BFS)
    if node ∈ next hierarchical level of nodes then
        DV ← update(DV, node.constraint.attribute);
    end if
    if node.constraint.attribute ∉ DV then
        DV ← D(id, node.constraint.attribute) ⋈id DV;
    end if
    NA ← computeAggregation(DV, VA, node.constraint.predicate);
    AR.add(NA);
end for
return AR;
```

TLAQ: Use Case IMMUCARE

To clearly show and present the benefits of performing analytical queries with *TLAQ*, we take the case of the *IMMUCARE* dataset. *IMMUCARE* is

a dataset storing data on cancer patients treated by immunotherapy, made available by the QUALITOP consortium members.

In this case, we focus on the female population and count their occurrences within the dataset. The root node is split into two nodes; the left side will be responsible for counting the number of entries where the *Date* is comprised between the years 2018 and 2019, while the right side deals with the same attribute for the period between 2020 and 2021 based on the *COUNT* predicate. We further investigate each branch with additional constraints, specifically with the use of the *SUM* predicate over the *Chemotherapy Dose* attribute over a range of values between 50 and 600 and by using the predicate *COUNT* the records over *White Blood Cells* attribute while accounting for the values comprised between 4.0 and 5.5 for the left and right branches, respectively. These last two constraints are further investigated through the attributes *ALAT* and *ASAT* from one side and *Creatinine* and *Glucose* from the other side using the *AVG* and *MIN* predicates, respectively.

Figure 5.13 shows an example of a *TLAQ* query over the *IMMUCARE* dataset.

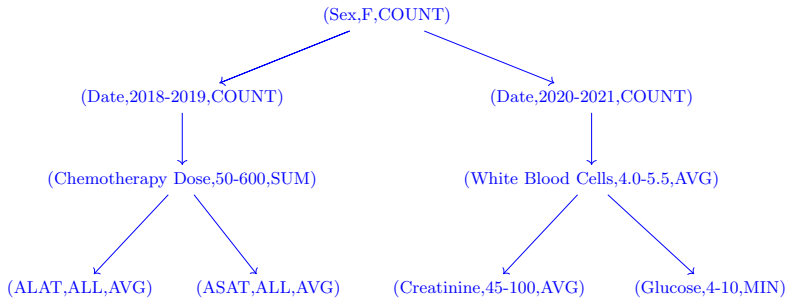


Fig. 4.12: TLAQ Example over the *IMMUCARE* Dataset

Selected Visualization Attributes

Table 4.1 shows the visualization attributes we used for the TLAQ query on the *IMMUCARE* dataset. Based on this setting, the TLAQ answer will be able to visualize analytics related to these attributes. Specifically, a data distribution related to each attribute and linked to each TLAQ answer node, will be accessible for presentation as data insight.

Root Node

In order to demonstrate and present the underlying processing that a constraint implies for each of TLAQ nodes, we take the root node of the TLAQ

Table 4.1: Visualization Attributes

Selection of Visualization Attributes	
Date	Chemo Tried Line Start
Age	Chemo Tried Line End Date
Chemo Tried Line	Concomitant Cancer Loc. Incl Cle
Chemo Tried Line Change	Concomitant Cancer Loc. Incl/Code

example on the *IMMUCARE* Dataset, where the constraint consists of (*Sex, F, COUNT*). After that, we provide an *SQL*-based query conversion that takes into consideration the visualization attributes for which we want to perform the analytics. The results for this node will then be represented as an aggregate value and a distribution plot, which is created based on visualization attributes. An illustration of the TLAQ analytics results for the root node is depicted in Figure 4.13.

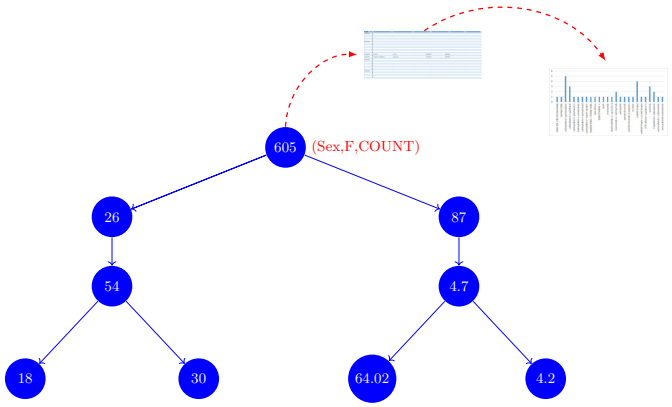


Fig. 4.13: TLAQ Analytics for Root Node

SQL Code for Creating the View. Figure 4.14 represents the *SQL* code used to create the view on the root node.
SQL Code for Generating Aggregate Result. Figure 4.15 shows the *SQL* code used to create the aggregate value on the root node, yielding the value of 605, which represents, in this case, the number of females in the dataset.
SQL Code for Generating Aggregate Distribution Plot for Attribute Tried Chemo Lines Incl/Chemo Tried Line. Figure 4.16 displays the *SQL* code used to create the aggregate distribution plot of the root node based on the visualization attributes. More into detail, we group-by a specific visualization attribute of the corresponding view (i.e., Tried Chemo Lines Incl/Chemo Tried Line).

```
-- Root node
CREATE VIEW Dataset-Q0(ID, Date, Age, [Tried Chemo Lines Incl/Chemo
    Tried Line], [Tried Chemo Lines Incl/Chemo Line Tried Line Change], [Tried
    Chemo Lines Incl/Chemo Line Tried Start Date], [Tried Chemo Lines
    Incl/Chemo Line Tried End Date], [Concomitant Cancer Localisation Incl
    Cle], [Concomitant Cancer Localisation Incl/Code]) AS
(SELECT ID, Date, Age, [Tried Chemo Lines Incl/Chemo Tried Line], [Tried
    Chemo Lines Incl/Chemo Line Tried Line Change], [Tried Chemo Lines
    Incl/Chemo Line Tried Start Date], [Tried Chemo Lines Incl/Chemo Line
    Tried End
Date], [Concomitant Cancer Localisation Incl Cle], [Concomitant Cancer
    Localisation Incl/Code]
FROM IMMUCARE
WHERE Sex = F;);
```

Fig. 4.14: Creating the View on the Root Node

```
-- Root node
SELECT COUNT(*)
FROM Dataset-Q0;
```

Fig. 4.15: Generating Aggregate Value on the Root Node

```
-- Root node
SELECT [Tried Chemo Lines Incl/Chemo Tried Line], COUNT([Tried Chemo
    Lines Incl/Chemo Tried Line])
FROM Dataset-Q0
GROUP BY [Tried Chemo Lines Incl/Chemo Tried Line];
```

Fig. 4.16: Generating Aggregate Distribution Plot

It is worth noting that in the previous analytics examples, the *JOIN* operation was not necessary on the views since the firstly created view, *Dataset-Q0*, includes all of the attributes required to create the subsequent views, namely *Dataset-Q1-1* and *Dataset-Q1-2* views.

Left Child Node of Root Node

Translating the left root child node constraint, namely: *(Date,2018-2019,COUNT)*, into *SQL* code aiming at creating the view needed to derive the aggregate and to obtain the distribution plot, the following *SQL* scripts are adopted to generate the TLAQ analytics results depicted in Figure 4.17.

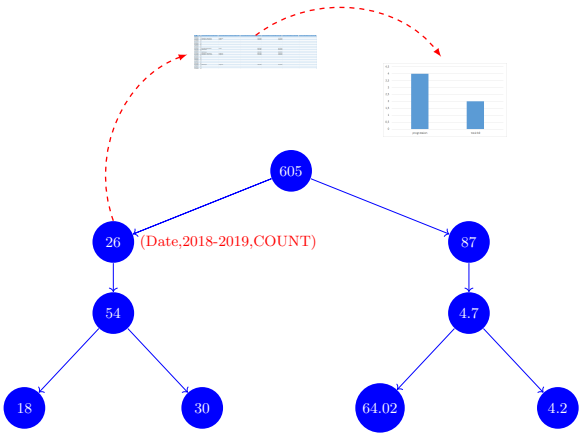


Fig. 4.17: TLAQ Analytics for Root Node Left Child

SQL for Creating the View. Figure 4.18 shows the *SQL* code used to create the view on the left child node of root node based on the visualization attributes.

```
-- Left child of root node
CREATE VIEW Dataset-Q1-1(ID, Date, Age, [Tried Chemo Lines Incl/Chemo
    Tried Line], [Tried Chemo Lines Incl/Chemo Line Tried Line Change], [Tried
    Chemo Lines Incl/Chemo Line Tried Start Date], [Tried Chemo Lines
    Incl/Chemo Line Tried End Date], [Concomitant Cancer Localisation Incl
    Cle], [Concomitant Cancer Localisation Incl/Code]) AS
(SELECT ID, Date, Age, [Tried Chemo Lines Incl/Chemo Tried Line], [Tried
    Chemo Lines Incl/Chemo Line Tried Line Change], [Tried Chemo Lines
    Incl/Chemo Line Tried Start Date], [Tried Chemo Lines Incl/Chemo Line
    Tried End Date], [Concomitant Cancer Localisation Incl Cle], [Concomitant
    Cancer Localisation Incl/Code]
FROM Dataset-Q0
WHERE Date BETWEEN #1/1/2018# AND #12/31/2019#;);
```

Fig. 4.18: Example of SQL Code Generating Views over TLAQ Nodes

SQL for Generating Aggregate Result. Figure 4.19 presents the *SQL* code used to create the aggregate value on the left child node of root node. The resulting value is 26, representing the count of female patients that underwent the cancer treatment between the years 2018 and 2020.

```
-- Left child of root node
SELECT COUNT(*)
FROM Dataset-Q1-1;
```

Fig. 4.19: SQL Query Generating Aggregate

SQL for Generating Aggregate Distribution Plot for Attribute Tried Chemo Lines Incl/Chemo Tried Line. Figure 4.20 displays the *SQL* code used to create the aggregate distribution plot on the root node based on the visualization attributes.

```
-- Left child of root node
SELECT [Tried Chemo Lines Incl/Chemo Line Tried Line Change],
    COUNT([Tried Chemo Lines Incl/Chemo Line Tried Line Change])
FROM Dataset-Q1-1
GROUP BY [Tried Chemo Lines Incl/Chemo Line Tried Line Change];
```

Fig. 4.20: Generating Aggregate Distribution Plot

Right Child Node of Root Node

Similarly, to what is done with the left child node, we translate the right child node of the root node, namely: *(Date,2020-2021,COUNT)*, into *SQL* code that aims at creating the necessary view for obtaining the aggregate value and the distribution plot. The following SQL scripts are adopted to generate the TLAQ analytics results shown in Figure 4.21.

SQL for Creating the View. Figure 4.22 is the *SQL* code used to create the view on the right child node of root node based on the visualization attributes.

SQL for Generating Aggregate Result. Figure 4.23 is the *SQL* code used to evaluate the aggregate on the right child node of root node based on the visualization attributes. The resulting value is 87, representing the count of female patients that underwent the cancer treatment between the years 2020 and 2021.

SQL for Generating Aggregate Distribution Plot for Attribute Tried Chemo Lines Incl/Chemo Tried Line. Figure 4.24 is the *SQL* code used to create the aggregate distribution plot on the right child node of the root node based on the visualization attributes. We could then visualize a histogram of counts for the values related to the attribute *Tried Chemo Lines Incl/Chemo Line Tried Line Change* of the *IMMUCARE* dataset.

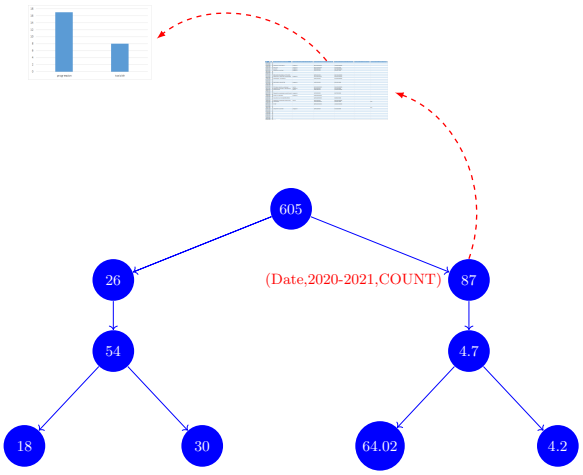


Fig. 4.21: TLAQ analytics for Root Node Right Child

```
—— Right child of root node
CREATE VIEW Dataset—Q1—2(ID, Date, Age, [Tried Chemo Lines Incl/Chemo
    Tried Line], [Tried Chemo Lines Incl/Chemo Line Tried Line Change], [Tried
    Chemo Lines Incl/Chemo Line Tried Start Date], [Tried Chemo Lines
    Incl/Chemo Line Tried End Date], [Concomitant Cancer Localisation Incl
    Cle], [Concomitant Cancer Localisation Incl/Code]) AS
(SELECT ID, Date, Age, [Tried Chemo Lines Incl/Chemo Tried Line], [Tried
    Chemo Lines Incl/Chemo Line Tried Line Change], [Tried Chemo Lines
    Incl/Chemo Line Tried Start Date], [Tried Chemo Lines Incl/Chemo Line
    Tried End Date], [Concomitant Cancer Localisation Incl Cle], [Concomitant
    Cancer Localisation Incl/Code]
FROM Dataset—Q0
WHERE Date BETWEEN #1/1/2020# AND #12/31/2021#;);
```

Fig. 4.22: SQL for Creating the View

```
-- Right child of root node
SELECT COUNT(*)
FROM Dataset-Q1-2;
```

Fig. 4.23: SQL Query Generating Aggregate

```
-- Right child of root node
SELECT [Tried Chemo Lines Incl/Chemo Line Tried Line Change],
    COUNT([Tried Chemo Lines Incl/Chemo Line Tried Line Change])
FROM Dataset-Q1-2
GROUP BY [Tried Chemo Lines Incl/Chemo Line Tried Line Change];
```

Fig. 4.24: Generating Aggregate Distribution Plot

4.4.2 Towards QFLS Predictive Analytics

The main purpose of designing such analysis tools is to enable useful data insights for medical recommendations. The most expressive way to display insights is through dashboards that group plots and statistics together under the same interface. Based on that, we decide to go for a dashboard that derives statistics on the returned TLAQ answers. For that purpose, we employ an *approximation formula* to generate a two-dimensional array of values based on the returned analytics. Specifically, we use the last level of analytics of each TLAQ answer the above-mentioned array serving as a baseline for our dashboard statistics. In more detail, given two TLAQ answers, $TLAQ_i$ and $TLAQ_j$, we define Equation (4.2):

$$A = \{a_i \mid a_i \in Leaves(TLAQ_i)\}_{i \in [1,n_i]}$$
$$B = \{b_j \mid b_j \in Leaves(TLAQ_j)\}_{j \in [1,n_j]} \quad (4.2)$$

where n_i, n_j are the breadth of $TLAQ_i$ and $TLAQ_j$ answers, respectively.

After that, we perform a *cross-computation* of the elements of A against those of B to generate a two-dimensional array V consisting of a two-dimensional summary of the TLAQ answers. V is defined through Equation (4.3):

$$V[i,j] = \left\lfloor \frac{a_i}{\sum_{i=0}^{n_i-1} a_i} \times \frac{b_j}{\sum_{j=0}^{n_j-1} b_j} + \frac{\sum_{i=0}^{n_i-1} a_i + \sum_{j=0}^{n_j-1} b_j}{n_i + n_j} \right\rfloor_{a_i \in A,\ b_j \in B} \quad (4.3)$$

Furthermore, we take the example of analytics depicted in Figure 4.32 and Figure 4.33 for which the sets A and B are shown in Figure 4.25:

By applying Equation (4.3) over the sets A and B, we then obtain the two-dimensional TLAQ answer array V shown in Figure 4.26, from which all dashboard plots are derived:

However, despite the fact that in our current implementation we considered the two-dimensional case, our approach is amenable to multidimensional cases through flattening techniques such as the one described in [55].

	0	1	2	3	4	5	6	7	8	9
A	14	330	1380	89	1613	3	58	34	18	58

B	16	327	1414	78	1638	0	46	18	14	46

Fig. 4.25: Arrays of Leaf Values for TLAQ $TLAQ_i$ (A) and $TLAQ_j$ (B)

V	328	560	23	3	298	0	6	2	1	4
	547	565	24	4	281	0	6	2	1	4
	26	28	1	0	15	0	0	0	0	0
	3	3	0	0	2	0	0	0	0	0
	294	282	13	2	153	0	3	1	0	2
	0	0	0	0	0	0	0	0	0	0
	7	7	0	0	4	0	0	0	0	0
	4	4	0	0	2	0	0	0	0	0
	1	1	0	0	1	0	0	0	0	0
	5	5	0	0	3	0	0	0	0	0

Fig. 4.26: Two-Dimensional TLAQ Answer Array V Generated from the TLAQ $TLAQ_i$ and $TLAQ_j$

4.4.3 QADAT at Work

In this Section, we delve into the working methodology of the QADAT tool, meticulously outlining its operational capabilities. We further elucidate its practical user interactions through detailed use cases and sequence diagrams.

QADAT provides available statistics about the content of the federated nodes for the user, specifically a histogram displaying the count of datasets in each federated node. In order to perform this process, information are collected from an XML *FederatedSystem.ini* file located within HDFS, which is parsed accordingly to extract required information (e.g., the dataset names or their IP locations). In summary, *QADAT* operates on a server application outside of the cloud cluster and communicates with the federated system in a stub-skeleton and client-server way through *RMI*-enabled method calls to submit requests and retrieve results to be shown inside the user interface. For what concerns the implementation of the *QADAT* client, it was achieved using *Spring v.* 5 *framework (Java v.* 17*)* and also using several JavaScript libraries such as *JQuery (v.* 3.6*)* for the dynamic rendering of the features, *ChartJs (v.* 3.7*)* and *Plotly (v.* 2*)* for the plot creation, and finally the graph visualization library *mxGraph (v.* 4.2*)* [19] for designing a graph drawing board (TLAQ editor).

Data Federation Discoverer Module (Federated System Explorer)

The data federation discoverer module represents the module that allows the user to select several datasets using a *tree-view component* supplied by the previously described configuration file. This component displays the node name, network IP address, and dataset hierarchy. After choosing a dataset name from the tree-view component hierarchy, the user is provided with an analysis of the anonymized dataset, such as its schema (e.g., type and attribute information) or a statistical summary (e.g., values average, percentiles). However, since complete data access is not permitted, the analysis contains a data sample with an excerpts of the actual dataset. Figure 4.27 shows the related DFD module interface.

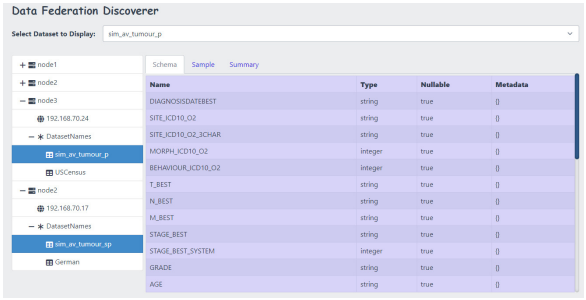


Fig. 4.27: DFD Module Interface

Anonymized Dataset Analysis Module (Metadata Analysis)

After performing the selection phase of datasets, which is enabled through the DFD module, the user can request a full metadata analysis of each selected dataset. The metadata analysis consists of an anonymization analysis that mainly informs about the degree of anonymization applied to each dataset, their total anonymization percentage information, the highest and lowest anonymization percentages, as well as the total record count. The metadata analysis provides attribute-related information such as its name, the type of the attribute, and their range of values, specifically the min and max values (the categorical attribute range is based on the alphabetical order). The corresponding anonymized dataset analysis module interface is presented in Figure 4.28.

Anonymized Dataset Analytics Environment Module (Data Analysis)

The anonymized dataset analytics environment module provides the user with a two-step process interface (enabled through *JQuery Steps* [186]). In the first

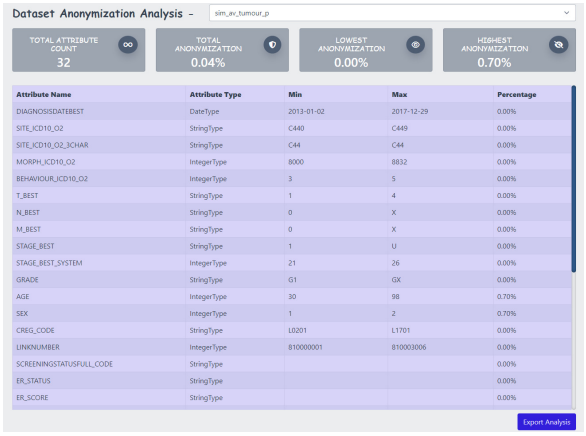


Fig. 4.28: ADA Module Interface

interface (see Figure 4.29), they are prompted with a *JQuery-based multi-select* component [49] to choose from the visualization attributes they need for analyzing their distribution against the TLAQ model. Then, in the second step, the user is provided with a graph drawing board through which they can compose and create the ($TLAQ$) structure (see Figure 4.31). The editing of the TLAQ nodes is made possible using a dialog box through which the user enters the constraint details, as shown in Figure 4.30.

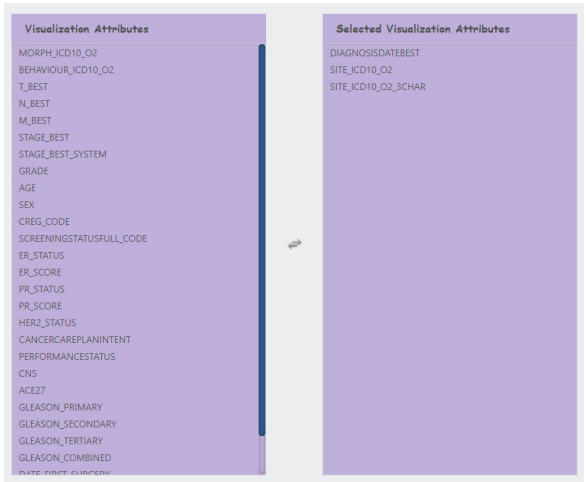


Fig. 4.29: Interface to Select Visualization Attributes

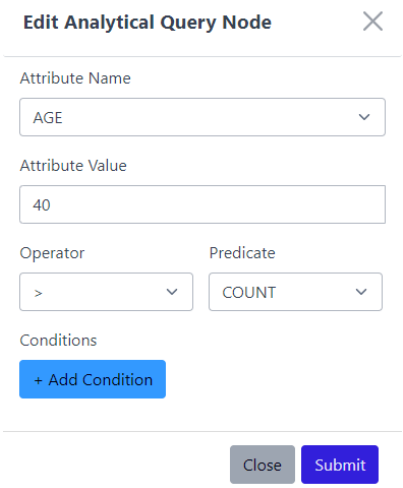


Fig. 4.30: Dialog Box to Edit a TLAQ Node Constraints

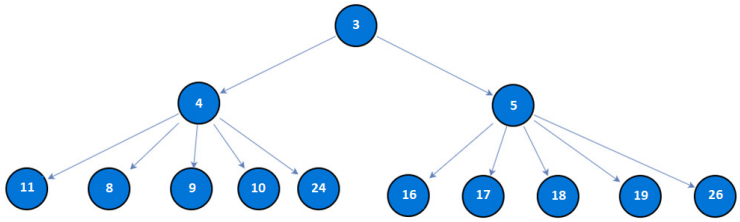


Fig. 4.31: TLAQ Example

The *TLAQ* composition is accomplished by first generating the tree-like graph and then specifying for each node a set of aggregate-operations-based constraints (e.g., Figure 4.31, a *COUNT* of records where $AGE > 40$ is specified as a constraint). The TLAQ could then be submitted simultaneously to all federated nodes involved in the analysis. The TLAQ answer is then obtained in a second window, where the user can display visualization attributes for distribution-related plots or inspect the nodes for the aggregated results for each of the analyzed datasets (by toggling the analysis by dataset). Results from executing the TLAQ (see Figure 4.31) are depicted in Figure 4.32 and Figure 4.33. Clearly, the aggregations drawn on each of the nodes are resulting from executing the associated constraints on the related data. Examples of distribution-based analytics are also shown in Figure 4.34. In addition to that, the system provides an import feature, consisting of a file chooser component, in order to import a previously created *TLAQ* using *XML* format. Note that a TLAQ export feature is made available as well in order to save, in *XML*

format, the structure of a created TLAQ. Exporting the full analytical results in PDF format is also possible using the export results functionality.

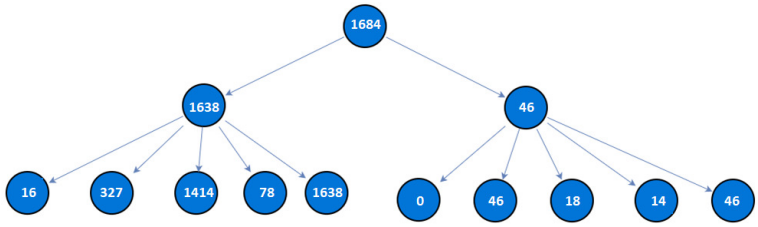


Fig. 4.32: TLAQ Answer Analytics 1

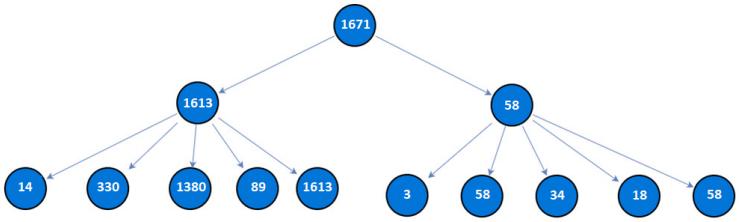


Fig. 4.33: TLAQ Answer Analytics 2

Predictive Analytical Environment (Dashboard)

As previously stated, a dashboard is available in our tool in order to enable recommendations based on the analyzed data. Analytics results provided by TLAQ execution are used to build a variety of visualizations. Figure 4.35 shows a visual representation of the dashboard plots.

As displayed in Figure 4.35, the following plots are available on this dashboard:

- *Aggregates*: obtained by computing mean values over rows of V and by grouping by attribute name;
- *Averages*: obtained by computing average values over rows of V;
- *Standard Deviations*: obtained by computing standard deviations over rows of V;
- *Clustering (Distance-Based)*: obtained by applying *k-means* clustering algorithm over rows of V;
- *Partition plot*: obtained through computing means for each attribute involved in V;

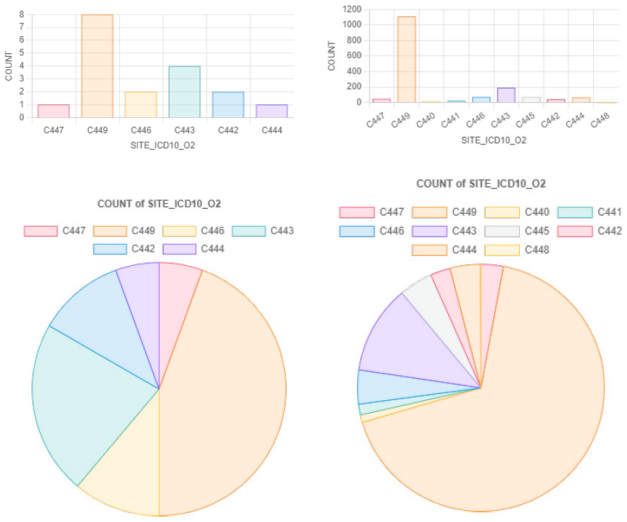


Fig. 4.34: Analytics on Distribution of Visualization Attributes

- *Correlation Matrix*: informing on the correlation levels among the considered attributes;
- *Pearson Correlation*: to measure Pearson correlation between the two query answers;
- *Spearman Correlation*: to measure Spearman correlation between the two query answers;
- *Outlier Detection (Density-Based Clustering)*: to detect outlier values.

Query History Exploration Module

Users (Data Analysts or Medical Staff) could fetch past TLAQ answers obtained through the *QADAT* client application. Query answers are referenced by the name of the submitter and the date of execution, and they are conveniently listed in an intuitive interface. They can also display a dynamic bar plot based on selected visualization attributes chosen at the moment of querying. Figure 4.36 presents the corresponding query history exploration module interface.

QADAT Use Case Diagram

The QADAT use case diagram provides a high-level overview of the tool functionalities and the interactions between users and the system. Figure 4.9 depicts a use case diagram for QADAT.

As displayed in Figure 4.37, a special attention goes to the differentiation between the two proposed user roles (i.e., data analyst and medical staff),

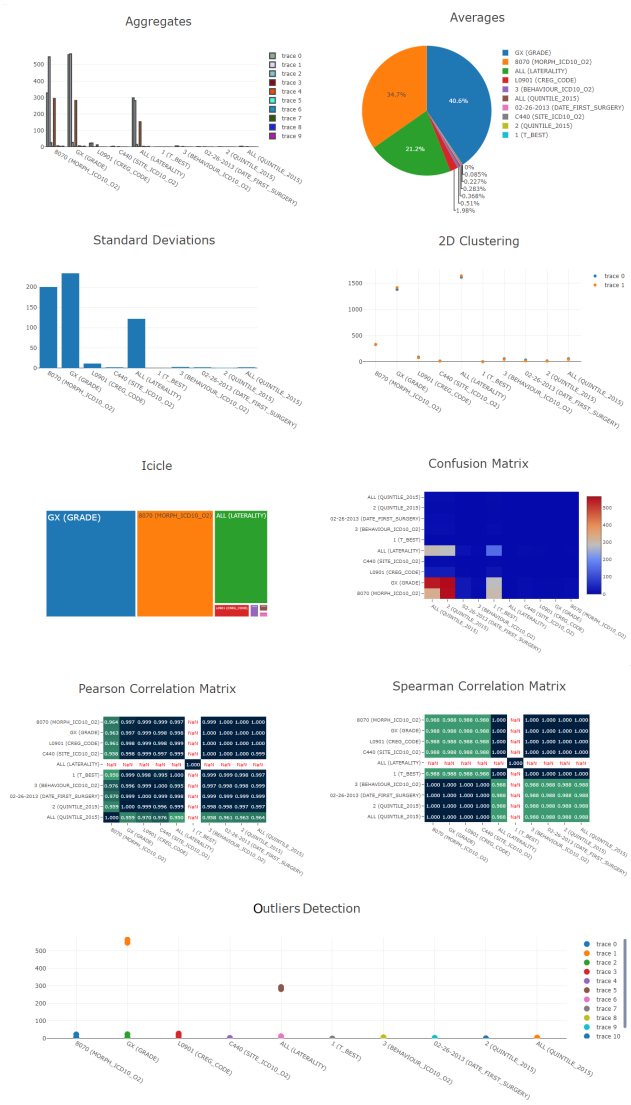


Fig. 4.35: Dashboard Plots

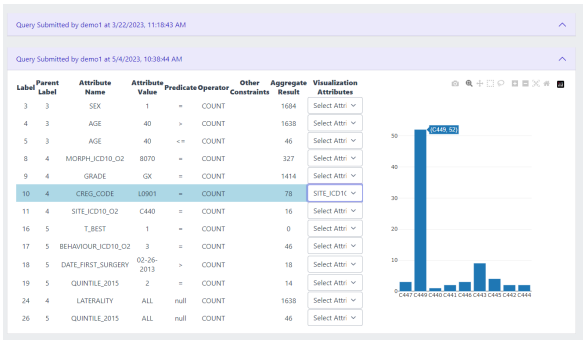


Fig. 4.36: Query History Exploration Module Interface

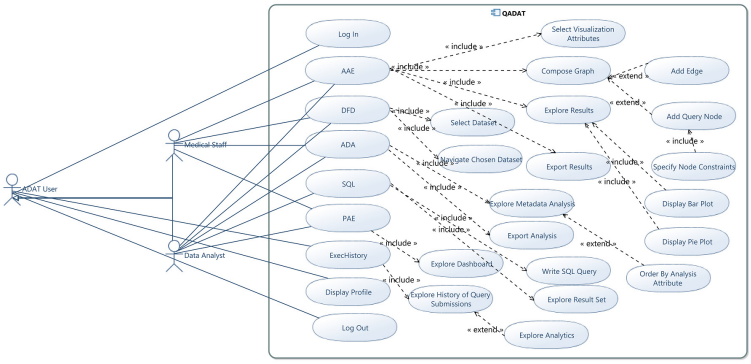


Fig. 4.37: QADAT Use Case Diagram

where both can access the following *DFD*, *ADA*, *AAE*, *PAE* modules and history of executions modules, but only the data analyst role has the permissions to access the *SQL* module.

QADAT Sequence Diagram

The sequence diagram serves to highlight the interaction between *QADAT* and both *RMI* server and the *Hadoop* instance. The corresponding sequence diagram for QADAT is shown in Figure 4.38.

As shown in Figure 4.38, Specifically, the interaction is performed between *QADAT* and both *RMI* server and the *Hadoop* instance for all the modules except the history execution module, which interacts only with the database to retrieve relevant historical data and with the *PAE* module to perform the computation needed for the plots locally.

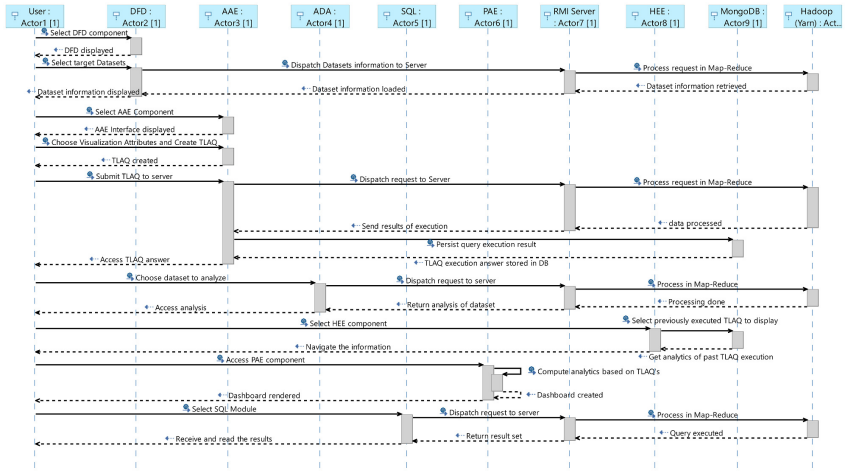


Fig. 4.38: QADAT Sequence Diagram

4.5 Case Study

In this Section, we provide a description of the innovative case study showing how the QFLS system operates and functions in practice over a real-life scenario. To further highlight the utility of our system and its practicality, we have developed a real-life scenario where two datasets are placed in two different locations (one in Portugal and another in Spain).

Furthermore, *Simulacrum* is a synthetic database that stores data about patients with cancer. Although being synthetic, *Simulacrum* maintains most of the properties of the original data with a high degree of accuracy. In addition, this database is a collection of linked clinical data tables available at *https://simulacrum.healthdatainsight.org.uk/*. Specifically, there are two main sets of tables, (*i*) Cancer Registration tables (*SIM_AV*); (*ii*) Systemic Anti-Cancer Therapy tables (*SIM_SACT*). Figure 4.39 presents the full schema of the *Simulacrum* database. This detailed representation provides a holistic understanding of the database structure.

As shown in Figure 4.39, this database contains a total of 8 tables, each encapsulating various attributes. However, the target table used in this experiment is the tumor table, which contains 2371281 rows and 32 columns/attributes (see Figure 4.42).

For the current experiment, we divided the mentioned dataset and used each part in different locations. The first one contains a dataset of 1048576 rows, and the second contains a dataset of 1048575 rows. Through QFLS cloud core, where 7 nodes were up, we executed the TLAQ query (see Figure 4.41) using the cluster mode of the YARN resource manager, and the results were then returned within roughly 2 minutes. For what concerns the details of the

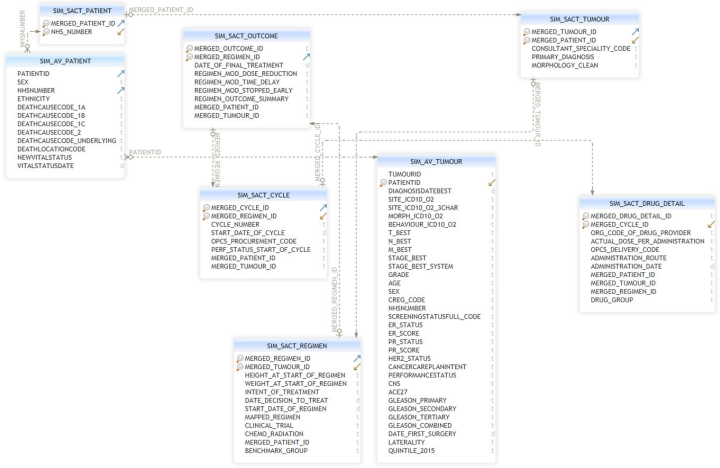


Fig. 4.39: Simulacrum Database Schema

Spark applications setting, for each application, (*i*) executor instances were set to 4; (*ii*) executor cores were set to 4; (*iii*) executor memory was set to $8GB$. Finally, the driver settings were left to default.

Figure 4.40 shows the full setting of the *Simulacrum Tumor* dataset on Spain and Portugal federated nodes.

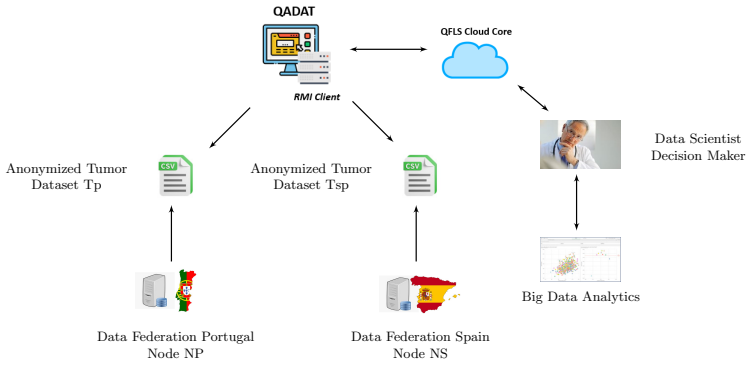


Fig. 4.40: Simulacrum Tumor Dataset on Two Federated Nodes

In this setting, and in order to derive recommendation-enabling insights over our selected datasets, we designed a TLAQ query, which is presented in Figure 4.41.

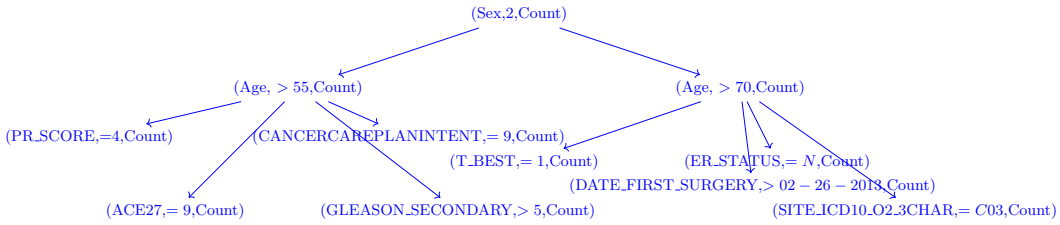


Fig. 4.41: TLAQ Used for Datasets Located in Spain and Portugal

As presented in Figure 4.41, we start by selecting the gender female, then we differentiate by age (greater than 70 or lower than 55). To further detail our query constraints and to obtain more granular data that will adequately serve our mathematical expression defined in Equation (4.2), we look to use specific attributes with valid values, such as $PR\_SCORE = 4$ or also $DATE\_FIRST\_SURGERY > 02 - 26 - 2013$.

Furthermore, the data used were taken directly from the *tumor.csv* dataset, available at *https://simulacrum.healthdatainsight.org.uk/available-data/table-descriptions/*. Figure 4.42 displays the schema of the tumor table with several attributes that reflect details about each registered tumor.

As shown in Figure 4.42, the following list of attributes characterizes the tumor table:

- *TumorId*: the tumor's unique identifier;
- *PatientId*: the patient's unique identifier;
- *DiagnosisDateBest*: age of the patient when diagnosed;
- *SiteIcd1002*: information about the site of the tumor, as character 4-ICD-10-O2 code;
- *SiteIcd10023char*: information about the site of the tumor, as 3-character ICD-10-O2 code;
- *MorphIcd1002*: information about the shape of the tumor;
- *BehaviorIcd1002*: information about the behavior of the tumor;
- *TBest*: information about the T stage of the tumor;
- *NBest*: information about the N stage of the tumor;
- *MBest*: information about the M stage of the tumor;
- *StageBest*: information about the best registry about the stage of the tumor;
- *StageBestSystem*: information about the system used to record the best registry about the stage of the tumor;
- *Grade*: grade of the tumor;
- *Age*: age of the patient when the tumor was diagnosed;
- *Sex*: sex of the patient;
- *CregCode*: code of the catchment area where the patient was resident when the tumor was diagnosed;

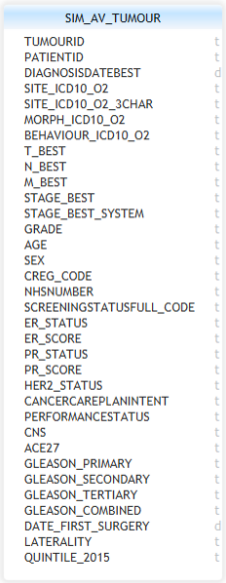


Fig. 4.42: Tumour Table Schema

- *LinkNumber*: identifier of the link to cross-dataset;
- *ScreeningStatusFullCode*: a code representing the full-detailed screening status of the tumor.

It is worth noting that, as a matter of data privacy, while processing data to extract the mentioned analytics, we suppressed every information linking a table record directly to a patient. Specifically, we discarded two direct identifier attributes of *SIM_AV_Tumor* (i.e., *TumorId* and *PatientId*). Using the TLAQ query (Figure 4.41), we obtain the two TLAQ analytics relative to Spain and Portugal queries, presented in Figure 4.43 and Figure 4.44, respectively.

Figure 4.45 shows the analytical dashboard of varying types of plots performed based on the previously distributed TLAQ answers in order to get the best recommendation for a specific group of cancer patients (e.g., those with an age greater than 70). It is worth noting that this Figure is the same of the Figure 4.35, where we shown the kinds of analytical plots supported by our system.

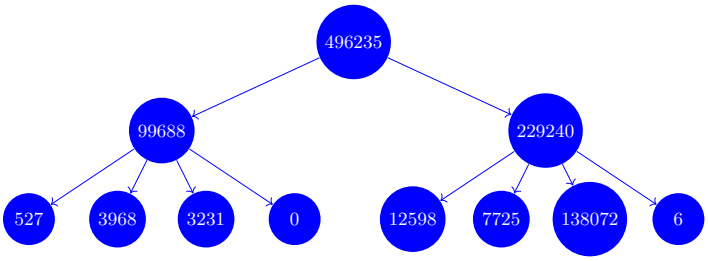


Fig. 4.43: TLAQ Analytics for Spain Dataset

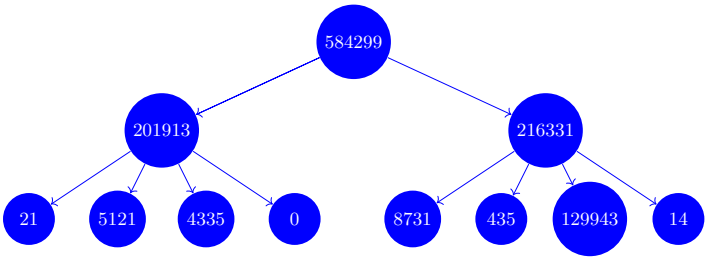


Fig. 4.44: TLAQ Analytics for Portugal Dataset

4.6 Summary

Through our QFLS system we employ a querying technique, namely the TLAQ query, to enable data analytics and derive a predictive dashboard useful for healthcare data analysis that enables medical guidelines and eventually recommendation for the patients. QFLS also includes capabilities such as anonymization analysis and statistical summarization of data located at each of geographically distributed nodes. In fact, QFLS leverages data federation to ensure data locality needed for the privacy of data in such environments. Indeed, data locality ensures that data are not transmitted back on forth over the system backbone network (preventing therefore privacy related attacks), but rather only aggregates will be computed in the distant nodes and returned to the main client application for enabling the analysis. Moreover, dealing with the data processing the way we propose, ensures that the disparate sources of data have specific anonymization techniques applied on them (through location-based data privacy guidelines and legislations) depending on the needs of the current node (and not one anonymization solution fits all) which will provide more privacy flexibility to the participating nodes and their associated data across the data federation system of QFLS. QFLS could also be considered as the building blocks to a data fabric that extends the capabilities of the state-of-the-art data lakes. In this sense, QFLS overcomes the disadvantages of centralized data lakes by potentially providing a domain specific analysis approach for the data that would lessen the limitations stemming from collecting the data and storing them into one single storage repository that further help in avoiding the well-known data swamp issue of data lake systems.

5

AB-DOM: An Algorithmic Framework for Supporting Privacy-Preserving Big Data Publishing in Big Data Lakes

In this chapter, we define our approach to anonymize healthcare data prior to use them for the analytical processes (e.g., through QFLS). We first, argue about the need to couple anonymity with diversity for improving the potential of privacy preserving data analytics processes. We, then; introduced and detailed, our framework with two working modes while we showed how it can be used over cloud settings through its user-friendly interface. Finally we assessed it considering a number of metrics while comparing it with other similar solutions. finally we highlight through proper theoretical arguments how ab-dom could be leveraged to data indexing ends in data lakes.

Given the sensitive nature of healthcare data, and in a matter of compliance with data protection and privacy regulations (GDPR – *General Data Protection Regulation*), there is a need to make data publishing more secure. In this context, we design and implement an innovative algorithmic framework called ***A****dvanced Privacy-Preserving* ***B****ig Data Publishing in Hierarchical* ***DOM****ains* (AB-DOM). AB-DOM is based on state-of-the-art anonymization techniques mixed with a *graph coloring algorithm* and an integrated *data sampling method* to guarantee that sensitive data are highly secured. Specifically, our proposed framework attempts at rising to the challenge of accounting for the diversity of patients while privacy-preserving their data for precision medicine in data lake environments. Indeed, with AB-DOM, it is proven that neither analytical results are diminished nor outlier cases are obliterated through the anonymization process. In a nutshell, AB-DOM enables effective and efficient data preserving data publishing in the context of *big data lakes* [51]. Broadly speaking, privacy preservation is at the heart of the data lakes where a critical step of the data ingestion phase is the anonymization of data. The latter step would ensure that data involved in the *analytical queries* are anonymous and are concealing well-enough sensitive patient information for example in the healthcare area. Other privacy related approaches of the data lake include data *pseudonymization* (e.g., [150]), *data encryption* (e.g., [147]), to mention a few. Additionally, we best describe our approach as "query-oriented" since the user (mainly, the competent medical

staff) has to specify, based on the target data source, a *hierarchy of domain values*, that is, basically, an *arrangement of constraints* that are pertinent for a specific analysis case. The latter step guarantees that the target subset of patients maximizes the value of the analysis outcome, and therefore enables quasi-optimal treatment recommendations. In order, to better support this, in our work we introduce the so-called *Tree-Like Analytical Queries* (TLAQ), a novel class of analytical queries that define a collection of *lazy hierarchical aggregations*, i.e. hierarchical aggregations that are not constrained to a *strict containment relationship*, over specific healthcare datasets, for analytics and decision-making purposes. As we will better describe in Section 5.2, analytical queries are capable of capturing typical *medical investigation processes*, which, more than ever, are occurring in *precision medicine* (e.g., [123]). Examples supporting this requirement are numerous . A pertinent example in oncology showing the utility of our solution is the following: let's suppose that in a population of cancer patients we want to preserve potential patients with a rare type of cancer for the analysis while anonymizing the data, for most existing solutions, patient entries with rare cancers would randomly be suppressed from the cohort made available to the analysis. Another example is in the context of the COVID-19 outbreak, where physicists were fully focused on isolating new variants in smaller collections of the target population, in order to progressively limit the virus's spread.

Furthermore, the precision medicine model embedded in AB-DOM exposes another important feature, i.e. the support for personalized data analysis by enabling cohort *diversity*. This concept, mixed with anonymization, has been introduced by the *DIVersity Anonymization* (DIVA) algorithm [138]. In AB-DOM's parlance, diversity refers to the fact that medical investigation should involve *small groups* which, classically, would be simply *suppressed* by the anonymization procedure, as highlighted by recent studies (e.g., [138]). This patient diversity-aware data analysis enhances the quality of healthcare analytics by accounting for *minority data values* and enabling precision medicine by targeting sub-groups of the patient population involved in the precision medicine process. To this end, we adopt DIVA as a core component for supporting diversity-aware privacy-preserving data publishing within the AB-DOM framework.

Contrary to this, standard literature anonymization techniques mostly focus on the type of re-identification attacks to prevent, while paying less attention to the diversity aspect of data that is, in most cases, omitted, in particular in the specific privacy-preserving healthcare analytics context. Indeed, much of minority-group data values would be obfuscated by classical approaches, thus resulting in entire (minority) groups of patients being discarded from the final analytics, and leaving no possibility for them to be considered for adjusted recommendations or treatments that would be suitable to their cases.

Seen from a real-life perspective, diversity would enable the representation of minority group data within the query/analytics, and therefore open the door to the analysis of all possible sub-groups of the entire considered data

Fig. 5.1: Precision Medicine Enabled through the AB-DOM Framework

domain. In this context, diversity fully supports precision medicine (see Figure 5.1), as mentioned before, given that minority groups can thus be captured by the target analytics, and, in turn, tailored treatment recommendations and actionable insights for each sub-group of the target patient population can be enabled.

Patients that are part of a small sub-group of people within the target dataset would then be recommended with a medical treatment that is very specific to their cases, despite the fact they account for a small percentage of the entire patient population. These same patients, under other diversity-unaware healthcare analytics frameworks, would then instead be recommended with the same treatment as the majority of other patients.

According to the evidences presented above, the AB-DOM framework should be considered among the first frameworks that exploit the idea of data diversity for the real-life problem represented by privacy-preserving healthcare analytics. In such a context, the AB-DOM framework also predicates the design of a flexible and intuitive user interface for practitioners that allow them to, firstly, access medical data in a safer way for patient critical information, hence for their *confidentiality* (e.g., [78]), and, secondly, provide the right amount of unconcealed data for optimal privacy-preserving data analytics, enriched by intuitive and facilitated *big data visualization* interfaces for an effective knowledge fruition. In addition to this, practitioners will also be able to query and analyze target privacy-preserving data and analytics, in order to receive actionable (privacy-preserving) insights.

Key Contributions

Key contributions, that the work described in this chapter are listed in the following:

- we provide the AB-DOM algorithmic framework for supporting privacy-preserving data publishing in the healthcare domain via anonymization, which has the merit of supporting diversity during the anonymization phase, i.e. avoiding to destroy small groups like other popular data anonymization techniques; this is a critical achievement for innovative precision medicine processes;

- we introduce a completely-novel analytical model implemented by the TLAQ queries that perfectly marry the typical (precision) medical investigation processes where medical operators incrementally analyze range of domains hierarchically connected by lazy aggregations;
- we provide a complete and detailed experimental evaluation and analysis of AB-DOM algorithm over both synthetic and real-life datasets, by proofing anonymization accuracy, temporal overhead and scalability of the algorithm.

Chapter Organization

The remaining part of this chapter is organized as follows. Section 5.1 illustrates the base algorithm enabling the diversity aspect in AB-DOM context. In Section 5.2, we highlight the main model we use to for AB-DOM. Section 5.3 provides a comprehensive description of the developed the AB-DOM algorithm with its two variants. After that, in Section 5.3.2, we present the user interface of AB-DOM while emphasizing on how it could be leveraged for entering the input and deriving the output. Finally, Section 5.5 focuses on the evaluation of the algorithm and depict the empirical results obtained from executing AB-DOM in a cloud environment.

Published Works

[59] Alfredo Cuzzocrea and Selim Soufargi. "An Algorithmic Framework for Supporting Privacy-Preserving Big Data Publishing in Big Data Lakes". In: *IEEE Transactions on Big Data* (2023). Currently under revision.

5.1 Preserving Diversity in Anonymized Data: The DIVA Algorithm

personal privacy is becoming a desideratum in many fields: a growingly pronounced need for considering the privacy of data in public publishing is now more real than ever (e.g., [148]). It is also a need turned into an incentive that has driven the adoption of new legislations concerning privacy-preserving in data publishing (e.g., [205]). Unfortunately, this significant rising only led to more anonymization techniques being proposed in the literature (e.g., [91]). On the other hand, while these techniques tightened the belt on the anonymization aspect of data management and analytics, they mostly missed on the diversity facet. Indeed, data, in all sorts of areas, very often include *minority attribute values* that current anonymization techniques fail to consider. This phenomenon can be seen in several application domains, ranging from the healthcare domain to the *e*-science domain, from the logistics domain to the domotic domain, and so forth.

DIVA [138] is an anonymization algorithm that accounts for the diversity of data, thus directly attacking the above-described research issue. In a step forward towards supporting more comprehensive accuracy in decision making, DIVA aims at representing minority groups as to prevent *bias* in data analytics. Indeed, most often, these minority groups are too restricted in number to be part of an analysis. Computational models may not "notice" their presence, and larger groups would dominate the input for the analytics.

DIVA is a *constraint-based framework* that unravels the challenge of data diversity in data anonymization. These constraints, which are part of the input for the DIVA algorithm, are expressed in terms of conditions on attribute values, considered to be part of small groups of attribute values within the target dataset. These conditions impose the level of representation of the perceived minority groups for the actual data analytics.

In more details, the input of DIVA is characterized by the following components: (i) a relation R to be anonymized; (ii) a set of diversity constraints Σ to be satisfied; (iii) a constant k that determines the degree of k-anonymity, according to the well-known k-anonymization procedure [188], which is the baseline anonymization technique of DIVA. The output of DIVA is represented by a Σ-diverse k-anonymous relation R'. The two principal processing steps of DIVA are the following ones: (i) the *clustering step*, where the input relation R is split into non-intersecting, k or greater sized, clusters based on the diversity constraint set Σ; (ii) the *suppression step*, which consists of concealing *quasi-identifier attribute values* [188] for clusters that do not meet the k-anonymous condition satisfied by the clustering step.

It is worth noting that, while the baseline version of DIVA makes use of k-anonymity as anonymization method, other state-of-the-art anonymization methods (e.g., *l-diversity*, *t-closeness*, and so forth) could be used as well.

DIVA algorithm is reported in Algorithm 2.

Algorithm 2 DIVA

Input: Dataset R, list of constraints Σ, integer k
Output: k-anonymous and diverse dataset

Begin
  $S_\Sigma \leftarrow diverseClustering(R, \Sigma, k)$;
  if $(S_\Sigma = \varnothing)$ **then**
    return null;
  end if
  $R_\Sigma \leftarrow suppress(S_\Sigma)$;
  for all $(C_i \in S_\Sigma)$ **do**
    $R \leftarrow R/C_i$;
  end for
  $R_k \leftarrow anonymize(R, k)$;
  return $integrate(R_\Sigma, R_k)$;
End

More specifically, DIVA is composed of four methods ensuring that the constraints covering the target minority groups are satisfied in the final anonymized dataset. In the following, we detail on these four methods:

- *diverseClustering*, which takes a set of constraints as input to create a *graph of attribute values*, colors its nodes whenever possible, and, based on that node coloring, returns the data of minority groups, denoted by S_Σ;
- *suppress*, which turns S_Σ into a *privacy-preserving complying dataset*;
- *anonymize*, which obfuscates attribute values of the initial dataset $R - S_\Sigma$, according to the adopted anonymization method (e.g., k-anonymity);
- *integrate*, combines S_Σ and the rest of the tuples from the original dataset R, and extracts a set of upper-bound constraints on the resulting the dataset – if, by integrating S_Σ with the rest of tuples from R, these upper-bound constraints are breached, then *Integrate* runs again as to make sure that the constraints are satisfied in the final anonymized dataset R'.

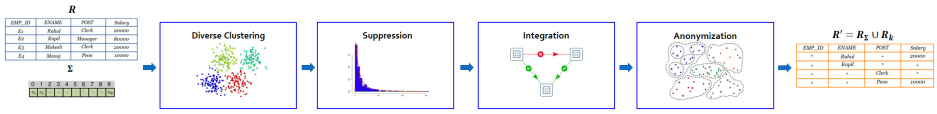


Fig. 5.2: The DIVA Algorithm Flow

From the theoretical analysis of the algorithm, it turns that time complexity of DIVA is *polynomial* with respect to the size of the input relation R. A summary of the DIVA algorithmic flow is depicted in the Figure 5.2.

In order to illustrate DIVA in action, we provide a concrete example on how DIVA works. To this end, from [138], we consider an input relation storing medical data with a slightly-modified version of the data schema, as to increase the expressive power. In particular, the original schema includes the following attributes: (i) *ID*, which models the identifier of the patient; (ii) *GEN*, which represents the gender of the patient; (iii) *ETH*, which models the ethnicity of the patient; (iv) *AGE*, which represents the age of the patient; (v) *PRV*, which models the province where the patient lives; (vi) *CTY*, which represents the city where the patient lives; (vii) *DIAG*, which models the diagnosis of the patient. In our slightly-modified version, we simply add a new attribute, called *S-ETH*, which models the sub-ethnicity of the patient. Figure 5.3 shows this simple yet effective schema variation.

Similarly, our modified input relation that stores medical records is reported in Table 5.1.

In DIVA, diversity constraints must be specified. These constraints specify: (i) the attribute name; (ii) the target attribute value of diversity; (iii) the

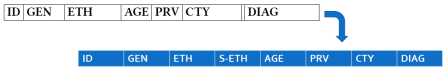


Fig. 5.3: Schema Variation of Input Medical Relation [138] of Table 5.1

Table 5.1: Input Medical Relation

ID	GEN	ETH	S-ETH	AGE	PRV	CTY	DIAG
r_1	Male	Caucasian	Western	80	AB	Boston	Hypertension
r_2	Male	Caucasian	Western	32	AB	Boston	Tuberculosis
r_3	Female	Caucasian	European	59	AB	Boston	Osteoarthritis
r_4	Female	Caucasian	White	46	MB	Dallas	Migraine
r_5	Female	African	Arabic	32	MB	Dallas	Hypertension
r_6	Female	African	Saharian	43	BC	New York	Seizure
r_7	Female	Caucasian	White	35	BC	New York	Hypertension
r_8	Male	Asian	Chinese	58	BC	New York	Seizure
r_9	Male	Asian	Chinese	63	MB	Dallas	Influenza
r_{10}	Male	Asian	Indian	71	BC	New York	Migraine

lower and upper bounds over the frequency of the specified attribute value. An example of these constraints as related to the input medical relation is the following:

$$\sigma = (CTY[NewYork], 2, 4) \tag{5.1}$$

where: (i) CTY is the attribute name; (ii) *New York* is the targeted attribute value; (iii) 2 and 4 are, respectively, the lower and upper bound over frequency of the event $CTY = New\ York$.

As in the original DIVA algorithm, an input diversity constraint set Σ must be set. To give an example, consider the following constraint set:

$$\begin{aligned}\Sigma = \{&\sigma_1 = (ETH[Asian], 2, 3),\\ &\sigma_2 = (ETH[African], 1, 3), \sigma_3 = (CTY[NewYork], 2, 4)\}\end{aligned} \tag{5.2}$$

Given the latter set Σ, the Naïve execution of DIVA returns in output the diversity-aware anonymized relation reported in Table 5.2.

Table 5.2: Diversity-Aware Anonymized Relation

ID	GEN	ETH	S-ETH	AGE	PRV	CTY	DIAG
p_1	Male	Caucasian	Western	*	AB	Boston	Hypertension
p_2	Male	Caucasian	Western	*	AB	Boston	Tuberculosis
p_3	Female	Caucasian	European	*	*	*	Osteoarthritis
p_4	Female	Caucasian	White	*	*	*	Migraine
p_5	Female	African	Arabic	*	*	*	Hypertension
p_6	Female	African	Saharian	*	*	*	Seizure
p_7	*	*	*	*	BC	New York	Hypertension
p_8	*	*	*	*	BC	New York	Seizure
p_9	Male	Asian	Chinese	*	*	*	Influenza
p_{10}	Male	Asian	Indian	*	*	*	Migraine

By inspecting the output relation in Table 5.2, it should be noticed that the so-obtained diversity-aware anonymized relation is *2-anonymous* (k-

anonymity is used as base anonymization method), while the diversity constraints satisfied and the minority groups well represented.

As shown in this Section, DIVA computes a privacy-preserving output dataset that satisfies a given set of diversity constraints, which, in many different ways, could prove to be useful in several research areas, especially healthcare analytics as it turns to be clearer and clearer. Despite this, a major drawback of DIVA is that it is geared towards *classical data architectures* (e.g., standalone data center) rather than big data lake architectures. Therefore, in order the diversity-aware privacy-preserving anonymization method to be used in the context of big data lakes, the DIVA algorithm must be adapted and integrated accordingly as to run in *Cloud-based high-performance big data computing platforms* (e.g., [183]).

5.2 Advanced Privacy-Preserving Data Publishing in Healthcare Analytics: The Tree-Like Analytical Query Model

The advent of new technologies in the healthcare domain has generated an unprecedented high amount of *big healthcare data*. These data are more and more subject to meaningful analytics, given their potential value. Indeed, mining these data would yield precious information that could revolutionize medicine, by allowing the discovery of unknown remedies for diseases, or more simply, saving lives (e.g., [72, 157]).

Privacy-preserving methods are somewhat complex to use by non-data-science-expert operators, since they require domain-specific knowledge (e.g., [140, 139]). The intervention of qualified personnel is indeed necessary in this case. This raises the problem of day-to-day use of these advanced techniques, since there is a need for extensively employing privacy-preserving data publishing on large volumes of publicly-shared data, for healthcare analytics purposes.

More specifically, privacy-preserving algorithms require domain-specific knowledge in order to be tuned for patient data in the specific healthcare analytics context. Most medical-staff/caretakers do not possess the required expertise to apply these algorithms, case-wise, on the patient records. Worse yet, they could be computer software-illiterates, and could not ever handle simple software tools unless they are really intuitive to use, and, in addition to this, equipped with a user-friendly interface.

Nevertheless, medical operators are experts in conceptually modeling queries for analytical tasks, based on a *declarative approach*. In this context, a need to come up with an easy-to-use way for medical staff to analyze patient data could be instrumental. Based on this assumption, it is of primordial concern to set up a declarative model for practitioners and other medical staff to query and analyze the data. *Declarative semantics* (e.g., [101]) would facilitate analytics over data for practitioners. Based on these motivations, in

AB-DOM the TLAQ model is introduced and experimentally assessed. Basically, a TLAQ is an intelligent manner to conceive a declarative semantics for expressing *medical investigation processes* in a *hierarchical fashion*. According to the TLAQ analytical model, medical operators are not constrained to set-up difficult parameters of privacy-preserving algorithms, but they only need to *graphically* edit TLAQ that are then used by the AB-DOM framework as input for returning privacy-preserving anonymized datasets, still organized according to a *tree-like indexing data structure*.

It should be noted here that such kind of medical investigation is completely coherent with the general precision medicine goal, as well as with the common practice of medical operators used for investigation. To become convinced of this, it suffices to think of the actual *COVID-19 outbreak*. Here, for instance, medical operators could be interested in investigating the number of COVID-19 cases in *North Italy* macro-region first and, within this domain, detailing the analysis in the hierarchically-contained domain represented by the *Lombardy* region.

Indeed, healthcare data (of clinical trials) typically have an inherent hierarchical structure because of the way they are captured. Based on that fact, it is practical to structure the data hierarchically to exploit it for analysis. In this sense, tree-like structures are clearly a useful and fashion-wise approach to build analytics queries over healthcare data. In addition to this, this way of structuring the data also enables *multi-level*, *multi-granular* personalized healthcare decision making, thus enriching the *expressive power* of the underlying analytics. Besides, this hierarchical-driven data analytics methodology allows medical operators to target exact records for extracting interesting knowledge and discovering hidden patterns in data.

In the TLAQ model proposed by AB-DOM, these innovative analytical queries are combined with the DIVA algorithm presented in Section 5.1, as to ensure diversity in the whole data anonymization procedure. Diversity is very relevant in healthcare analytics, according to what discussed in Section 5.1. Focus again on the running example of COVID-19 data analytics. Here, medical operators are usually interested even in detecting new virus variants that may occur in *small populations* of patients. The latter populations are just those preserved thanks to the diversity-aware anonymization methodology (see Section 5.1). It is worth to notice here that one of the main intuitions of the AB-DOM proposal is represented by the combination of the innovative TLAQ model with the diversity-aware anonymization technique proposed by DIVA algorithm.

Figure 5.4 shows an example TLAQ. Here, through a simple *graphical user interface*, medical operators edit a TLAQ on the medical dataset shown in Table 5.1. First, at the root node, medical operators are interested in accessing (in a privacy-preserving manner) the healthcare data of the patient population composed by people of Asian ethnicity having an occurrence frequency within the interval [5:55]. Then, at the first level of the TLAQ, *within the previous domain*, in a hierarchical fashion, medical operators are interested in access-

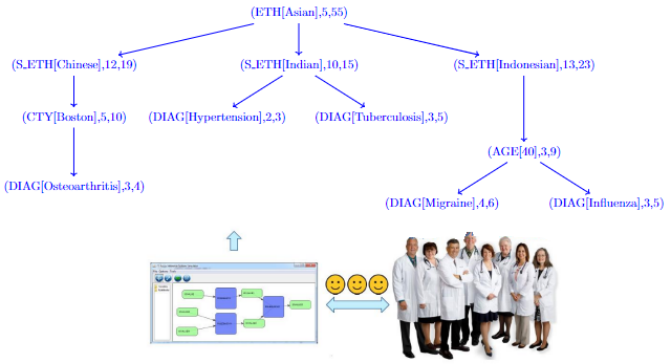


Fig. 5.4: Example TLAQ

ing (in a privacy-preserving manner) the healthcare data of *(1)* the patient population composed by people of Chinese sub-ethnicity (*within the Asian ethnicity*) having an occurrence frequency within the interval [12:19], *(2)* the patient population composed by people of Indian sub-ethnicity (*within the Asian ethnicity*) having an occurrence frequency within the interval [10:15], and, finally, *(3)* the patient population composed by people of Indonesian sub-ethnicity (*within the Asian ethnicity*) having an occurrence frequency within the interval [13:23], respectively. All the other node queries of the remaining levels of the TLAQ shown in Figure 5.4 are self-expressive, so that they do not need further explanation.

When a TLAQ is executed, it returns, for each node query, a diversity-aware privacy-preserving data domain that is thus ready for advanced healthcare analytics. Figure 5.5 shows the blueprint concept of such a process, having k-anonymity as base anonymization algorithm.

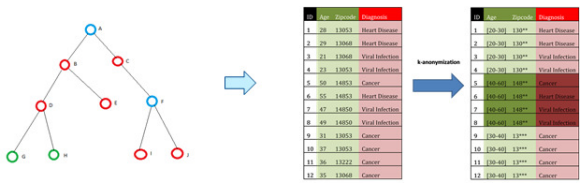


Fig. 5.5: Example TLAQ Execution with an Output Diversity-Aware Privacy-Preserving Data Domain (k-Anonymity is used as Base Anonymization Algorithm)

Looking into more implementation details, TLAQ are executed by AB-DOM algorithm over healthcare data in the Cloud, as highlighted in Section 5.1. This returns anonymized datasets that are still accessible via a corre-

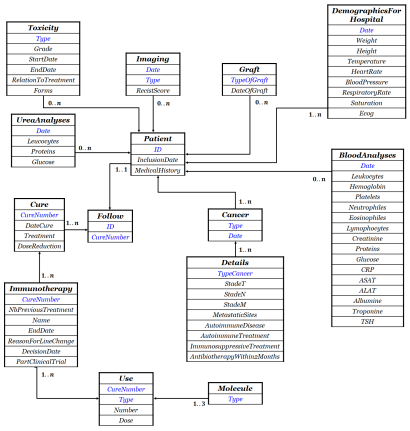


Fig. 5.6: IMMUCARE Data Schema

sponding tree-like data indexing structure. Finally, these anonymized datasets are accessed and managed by medical operators for data analytics purposes. Indeed, query evaluation over a Cloud-based backend is, at now, a very interesting research challenge, as pinpointed by recent studies.

Finally, the proposed TLAQ model can be further extended via innovative *deep learning approaches* (e.g., [8]), with the aim of effectively and efficiently analyzing healthcare data in a diverse and anonymized fashion. Indeed, intelligent and dynamic approaches, through *artificial intelligence*, could be used to automatically build hierarchies that serve as benchmarks for *automatically-creating analytical queries*. *Machine learning*, in fact, has already proved its worth in solving data analytics problems (e.g., [77, 178, 145]). Therefore, deep-learning-based algorithms could help the setting, in an automatic manner, of the hierarchy levels of a TLAQ to be submitted to the QUALITOP data lake. Ultimately, this would lead to lesser *human-based involvement* for analyzing the data, and would therefore increase effectiveness and accuracy of the final analytics.

5.2.1 TLAQ Model At Work: Examples Defined on Top of the IMMUCARE Dataset

QUALITOP makes use of a real-life dataset called *IMMUCARE*, provided by the partner consortium. IMMUCARE stores information about patients treated by immunotherapy against cancer, and all the related clinical examinations. Figure 5.6 shows the IMMUCARE data schema. Here, for instance, information about patients, their hospital demographics, their cancer illness along with the stage of the illness, their immunotherapy treatments (starting dates, duration of treatments, doses, etc.), their blood analysis, their urea analysis, their associated medical images, and so forth. It is worthy to notice

that all these data are of critical relevance for medical operators willing to support analytics over big healthcare data, yet in a privacy-preserving manner via AB-DOM.

Now we focus the attention on some TLAQ defined on top of the IMMUCARE dataset.

Figure 5.7 shows the first example TLAQ on IMMUCARE, named as $TLAQ_i$. Here, at the root node, $TLAQ_i$ targets the male patient population, by imposing an occurrence frequency within the interval [15:52]. Within this so-defined domain, $TLAQ_i$ then focuses on *(1)* patients having prostate cancer with an occurrence frequency within the interval [10:17], *(2)* patients having lung cancer with an occurrence frequency within the interval [11:17], *(3)* patients having liver with an occurrence frequency within the interval [14:23], respectively. All the other node queries of the remaining levels of $TLAQ_i$ shown in Figure 5.7 are self-expressive, so that they do not need further explanation.

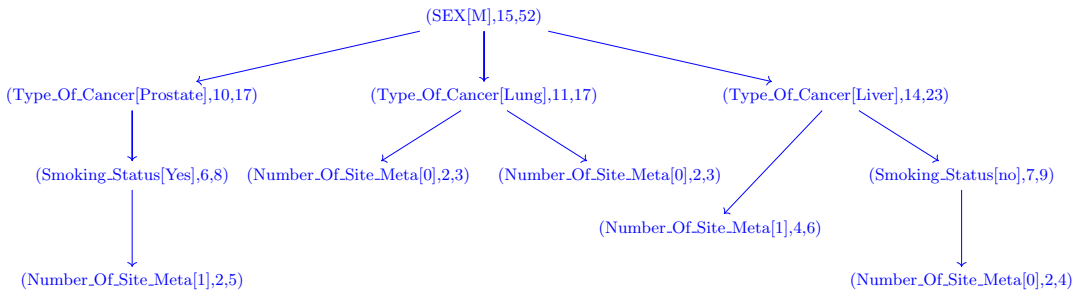


Fig. 5.7: TLAQ $TLAQ_i$

Figure 5.8 shows the second example TLAQ on IMMUCARE, named as $TLAQ_j$. Here, at the root node, $TLAQ_j$ targets the patient population having height equal to 180cm, by imposing an occurrence frequency within the interval [20:60]. Within this so-defined domain, $TLAQ_j$ then focuses on *(1)* patients having temperature equal to $36°C$ with an occurrence frequency within the interval [8:17], *(2)* patients having temperature equal to $37°C$ with an occurrence frequency within the interval [9:13], *(3)* patients having temperature equal to $38°C$ with an occurrence frequency within the interval [9:18], respectively. All the other node queries of the remaining levels of $TLAQ_j$ shown in Figure 5.8 are self-expressive, so that they do not need further explanation.

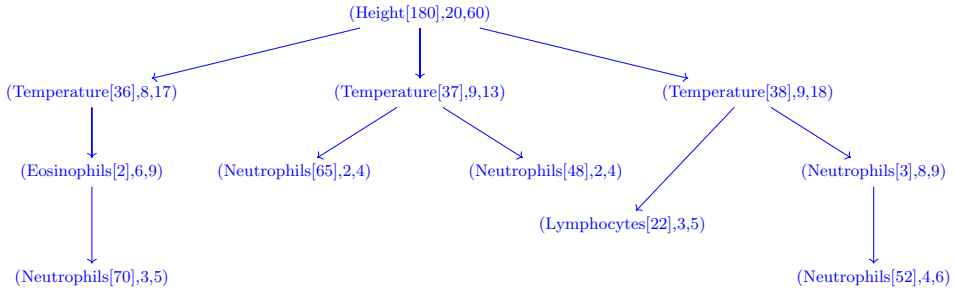


Fig. 5.8: TLAQ $TLAQ_j$

5.3 The AB-DOM Algorithm

AB-DOM is a framework whose main goal consists in coupling data privacy with data diversity, thus enabling precision medicine through healthcare analytics (e.g., [106]). In this Section, we focus on the AB-DOM algorithm in greater detail.

AB-DOM can execute in two modes: the *Naïve mode* and the *optimized mode*. The Naïve mode runs in a highly-demanding computational environment where relevant resources are available (typically, a big data lake in the Cloud). The optimization mode, still executing in a highly-demanding computational environment, aims at reducing the overall computational cost resulting from AB-DOM runs, via ad-hoc *optimized data-driven solutions*.

5.3.1 AB-DOM Naïve Mode

In the AB-DOM naïve mode, given the input TLAQ Q, each node query Q_i of Q is processed through the application of DIVA over the dataset associated to Q_i, named as *node dataset* R_i. In particular, at the root node Q_0, the associated dataset corresponds to the input dataset/relation R. At every internal and leaf node Q_j, the node dataset R_i is *inherited* from the parent node Q_i, being the latter the anonymized dataset R_i' obtained by executing DIVA on R_i at the node query Q_i. During the execution, Q is visited according to a *breadth-first strategy*, hence every node query, except the root node query, is evaluated on top of an anonymized dataset. Overall, throughout the entire process, every dataset associated to a node query is progressively characterized by an increased level of privacy-preservation. Figure 5.9 shows an execution example of AB-DOM naïve mode on top of the input relation storing medical records of Table 5.1, according to the schema variation of Figure 5.3.

As shown in Figure 5.9, the example TLAQ first considers a patient population with $ETHNICITY = Asian$, by imposing to have a sub-set of at least 5 records and at maximum 55 records. Then, within the so-

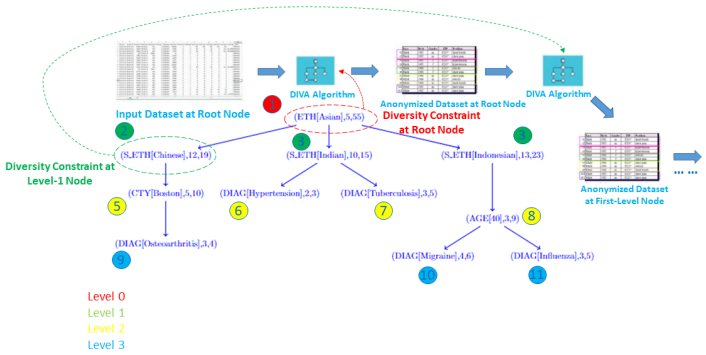


Fig. 5.9: AB-DOM Naïve Mode Execution Example

computed population, the overall precision medicine process further investigates, within the latter range, three different patient populations having, respectively, *SUB-ETHNICITY = Chinese, SUB-ETHNICITY = Indian*, and *SUB-ETHNICITY = Indonesian* (second level of the TLAQ). Further, within the range having *SUB-ETHNICITY = Chinese*, the population of people living in Boston is considered (*CITY = Boston*). Within the range having *SUB-ETHNICITY = Indian*, instead, the population of people that having been diagnosed with hypertension (*DIAG = Hypertension*) and tuberculosis (*DIAG = Tuberculosis*) is targeted. Finally, within the range having *SUB-ETHNICITY = Indonesian*, the population of people that having been diagnosed with migraine (*DIAG = Migraine*) and influenza (*DIAG = Influenza*) is addressed (third level of the TLAQ), and, within the latter range, those people being 40 years old (*AGE = 40*) (last level of the TLAQ).

Based on the described process, at the end of the evaluation of the input TLAQ Q of the actual AB-DOM naïve mode example shown in Figure 5.9, each node query Q_i of Q is equipped with a diversity-aware anonymized dataset R'_i (built from the user-defined diversity constraints), which healthcare operators, including physicists, and medical staff, are allowed to access and query in order to get insights and analytics about a specific segment of the population involved in the main precision medicine process.

In addition to this, anonymized datasets computed by AB-DOM can, of course, be used for supporting big data analytical tasks on top of the target healthcare domain. The latter tasks are meant for several purposes, for instance with the goal of supporting epidemiological decision-making like in the recent COVID-19 outbreak.

The output of the AB-DOM algorithm is a tree-like data structure, which, at the end, acts as an *indexing data structure*, where each (query-result) node links the reference diversity-aware anonymized dataset. Figure 5.10 shows an example of such an output data structure. It should be noted that this par-

ticular data structure is easily mapped onto a Cloud infrastructure where indexing nodes can be deployed over one or more Cloud nodes, and, in turn, (anonymized) datasets can be partitioned over multiple Cloud nodes, depending on their size and complexity.

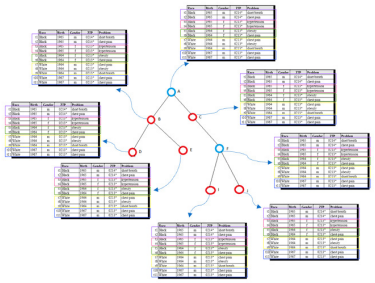


Fig. 5.10: Output AB-DOM Tree Like Data Structure with Anonymized Datasets

Finally, Algorithm 3 shows the AB-DOM naïve mode algorithm.

5.3.2 AB-DOM Optimized Mode

The main idea of AB-DOM is that of running in a high-performance Cloud computing setting with the goal of processing and supporting analytics on top of big healthcare datasets. Therefore, AB-DOM algorithm casts for high amounts of computational resources (space and time). As a consequence, we designed and developed the second AB-DOM mode, called AB-DOM optimized mode, whose main goal consists in achieving a more resource-aware and cost-efficient solution over the AB-DOM naïve mode (see Section 5.3.1). In this context, the optimized mode of AB-DOM comes as an alternative to reduce the overall computational cost needed from executing AB-DOM within a big data lake (like predicted by the main QUALITOP architecture).

In the optimized mode, AB-DOM works as follows. Given the input TLAQ Q, only the root node Q_0 of Q is processed through the application of DIVA over the dataset associated to Q_0, R_0. All the other internal nodes are processed by means of the *differential-privacy-based sampling technique* [83], in order to derive the node dataset R_i associated to the *children* node query Q_i of Q from the node dataset R_{i-1} associated to the *father* node query Q_{i-1} of Q. The final goal of this approach consists in achieving, in R_{i-1}, differential privacy anonymization (e.g., [82]) through sampling (e.g., [132]) the dataset R_i. Obviously, the sampling procedure allows us to reduce the overall computational cost with respect to the case of executing DIVA over each node query. Figure 5.11 shows an execution example of AB-DOM optimized mode

Algorithm 3 AB-DOM Naïve Mode

Input: Dataset D_0, Tree-Like Query T_Q
Output: Tree-Like Structure with Anonymized Datasets S_A

```
Begin
    S_A ← null;
    Queue Q ← null;
    Node N ← null;
    Dataset D ← null;
    DiversityConstraint DC ← null;
    AnonymizedDataset D_A ← null;
    Node childN ← null;
    if (T_Q ≠ null) then
        S_A ← new Tree();
        N ← T_Q.getRoot();
        DC ← N.getDiversityConstraint();
        D_A ← DIVA(D_0, DC);
        S_A.add(N, D_A);
        Q ← new Queue();
        childN ← N.getChild();
        while (Q.isNotEmpty()) do
            N ← Q.getRoot();
            Q.remove();
            D ← N.getDataset();
            DC ← N.getDiversityConstraint();
            D_A ← DIVA(D, DC);
            S_A.add(N, D_A);
            childN ← childN.getChild();
            while (childN ≠ null) do
                Q.add(childN);
                childN ← childN.getSibling();
            end while
        end while
    end if
return S_A;
End
```

on top of the input relation storing medical records of Table 5.1, according to the schema variation of Figure 5.3.

From Figure 5.11, since the underlying precision medicine process is the one described for the AB-DOM naïve mode case (see Figure 5.9), here we focus on algorithmic aspects only. As shown in Figure 5.11, DIVA is applied to the root node query Q_0 of the input TLAQ Q. Then, for all the other nodes, the diversity-aware anonymized datasets are obtained via applying the sampling-based (ϵ, δ)-differential privacy algorithm [83].

In more details, the latter technique [83] consists of applying *uniform sampling without replacement* (i.e., *reservoir sampling* – e.g., [196]), such that the

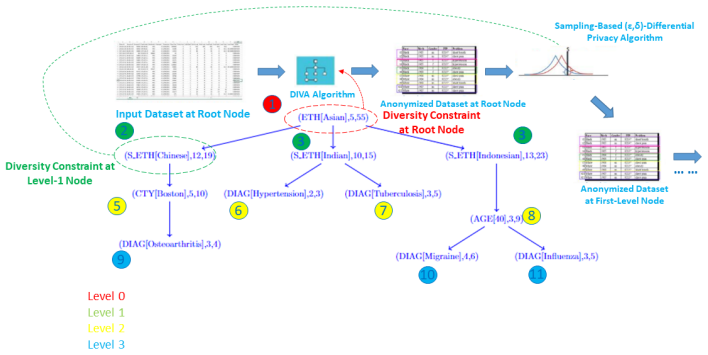


Fig. 5.11: AB-DOM Optimized Mode Execution Example

(ϵ, δ)-differential privacy condition [82] is satisfied. The benefits of using the latter approach are twofold. In fact, the sampling-based (ϵ, δ)-differential privacy algorithm ensures, from a side, an output anonymized dataset to be obtained and, from another side, a lower computational cost during the execution of AB-DOM can be achieved. In addition to this, in the AB-DOM optimized mode, the sampling-based (ϵ, δ)-differential privacy algorithm allows us better anonymized datasets to be obtained, due to the fact that (ϵ, δ)-differential privacy is, all considering, a more advanced anonymization technique than all variants of the DIVA algorithm employed in the Naïve mode.

Looking into details, the general differential privacy technique consists in the fundamental process of *data noise injection*, with the goal of ensuring anonymization. During this process, statistical properties of data are preserved, whereas specifics of individual records are concealed through the mentioned data perturbation method. Moving to more specific (ϵ, δ)-differential privacy, given two datasets R_i and R_j differing on at most one record, a *randomized function* K is able to achieve (ϵ, δ)-differential privacy for R_j over R_i if, for every sub-domain $S \subseteq Range(K)$, the following property holds:

$$Pr[K(R_i) \in S] \leq exp(\epsilon) \times Pr[K(R_j) \in S] + \delta \tag{5.3}$$

being $Pr[L_k]$ the probability distribution of the data domain L_k.

Another more practical definition of the (ϵ, δ)-differential privacy property is the following: two "adjacent datasets" R_i and R_j (i.e., datasets that differ by one record only) are $exp(\epsilon)$ and some residual noise δ indistinguishable if they verify the property. Indeed, it should be noted that, through this data anonymization technique, a patient could anonymously share her/his information for the big healthcare data analytics purposes, without fearing the hack or the disclosure of *personal data*. Not by chance, differential privacy is recognized as one of most reliable data anonymization methods [29].

In our specific AB-DOM optimized mode implementation, we use sampling to finally achieve the (ϵ, δ)-differential privacy property, so that, finally, we introduce a sampling-based (ϵ, δ)-differential privacy technique. Indeed, it should be noted that this approach defines a clever way to support scalability (i.e., reducing the overall complexity of accessing and processing big datasets) and targeting the final anonymization goal, within one single process. In the AB-DOM optimized mode, the sampling-based (ϵ, δ)-differential privacy technique is implemented by the *SampligDiffPrivacy* algorithm, shown in Algorithm 4.

Algorithm 4 Differential Privacy Sampling Algorithm

Input: Dataset D
Output: A Dataset Sample Satisfying Differential Privacy K

```
Begin
  for (i ← 1, |D|) do
    if (i ≤ n) then
      K[i] ← D[i];
    else
      r ← random(1, i);
      if (r ≤ n) then
        K[r] ← D[i];
      end if
    end if
  end forreturn K[1..n];
End
```

As mentioned, with the AB-DOM optimized mode, both goals are achieved: generating an anonymized (sampled) output dataset by also taming computational overheads of evaluating such algorithm on big healthcare datasets. Through our research, we demonstrate that AB-DOM optimized mode not only preserves the diversity of data (as required by precision medicine processes) but also the sampling phase meets the computational overhead reductions necessary within a big data lake environment. More into details, through the recursive iteration of the reservoir sampling algorithm, different diversity constraints can be progressively matched, with a certain degree of *flexibility*, by also finally generating anonymized datasets thanks to the differential privacy technique.

Like for the AB-DOM naïve mode (see Section 5.3.1) the output of the AB-DOM optimized mode is a tree-like data structure acting as an indexing data structure where each (query-result) node links the reference diversity-aware anonymized dataset (see Figure 5.10).

Finally, Algorithm 5 shows the AB-DOM optimized mode algorithm.

Algorithm 5 AB-DOM Optimized Mode

Input: Dataset D_0, Tree-Like Query T_Q
Output: Tree-Like Structure with Anonymized Datasets S_A

```
Begin
    S_A ← null;
    Queue Q ← null;
    Node N ← null;
    Dataset D ← null;
    DiversityConstraint DC ← null;
    AnonymizedDataset D_A ← null;
    Node childN ← null;
    if (T_Q ≠ null) then
        S_A ← new Tree();
        N ← T_Q.getRoot();
        DC ← N.getDiversityConstraint();
        D_A ← DIVA(D_0, DC);
        S_A.add(N, D_A);
        Q ← new Queue();
        childN ← N.getChild();
        Q.add(N);
        while (Q.isNotEmpty()) do
            N ← Q.getRoot();
            Q.remove();
            D ← N.getDataset();
            DC ← N.getDiversityConstraint();
            D_A ← samplingDiffPrivacy(D, DC, ϵ, δ);
            S_A.add(N, D_A);
            childN ← childN.getChild();
            while (childN ≠ null) do
                Q.add(childN);
                childN ← childN.getSibling();
            end while
        end while
    end if
return S_A;
End
```

5.4 AB-DOM in Action!

In order to make AB-DOM more appealing to use for practitionners, we designed a software tool to enable the control of the algorithm processing. In what follows we provide an example on the optimized mode and show AB-DOM in action.

AB-DOM front-end GUI was implemented using JavaFx 17 along with the help of SceneBuilder 17, and the backend was developed using Apache Spark v.3.3 and Java 17. A proper Hadoop cluster is available in premise to process the queries defined and submitted through the GUI. The workflow of the

application is described through the sequence diagram depicted in Figure 5.12.

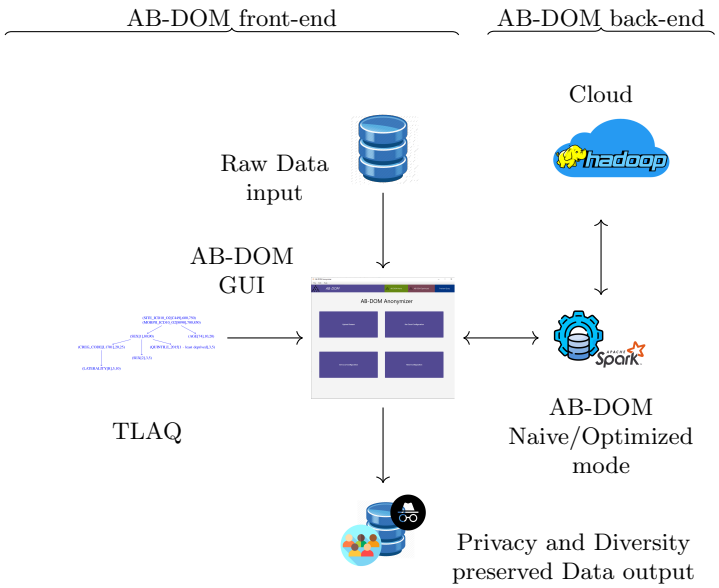


Fig. 5.12: The AB-DOM Workflow

A simplified workflow description of AB-DOM is illustrated in the figure 5.15. Firstly, the user is offered to upload a dataset. The tool features interfaces for the uploading of CSV/TSV/PSV or other types of dataset like JSON. In order to process the data accordingly (e.g., based on the TLAQ) The user could select either a local or a cloud mode of execution. For the cloud mode, the user will have to specify information such as the cloud mode type (standalone or YARN), the namenode URL or the memory and CPU resources to use for each of the executing datanodes. As shown in figure 5.16, after selecting the mode of execution we want to use for a TLAQ submission, we start by selecting QID attributes, then choosing different kind of options related to the anonymization process such as the transformation method (generalization or suppression) or also the privacy parameter (l or k for respectively, l-diversity or k-anonymity). To add that for the case of e.g., l-Diversity, the selection of sensitive attributes required by the anonymization process is necessary. In our tool, selecting such attributes is intuitive as a drag and drop action as shown in figure 5.17. In addition, our tool enables hierarchy based anonymization: by uploading a hieararchy file defining the hierarchical based conversions to be applied depending on the attribute values, the anonymization process yields values that are anonymized through generalization. Furthermore, and

in order to create a TLAQ query to submit over the data to anonymize, our tool proposes a straightforward way through creating a treeview component by specifying its nodes and its structure, or simply by uploading a JSON-formatted query through the dedicated functionality. An example of a JSON-based optimized mode translation of the TLAQ query depicted in Figure 5.13 is shown in Figure 5.14. At the end of the TLAQ creation process an option to visualize/preview the TLAQ structure is available for the user. Another useful functionality available in the GUI is the 'save query' option that offers the flexibility of saving a query created by the user (through the treeview component) for a later reload and use all in JSON format. A fact worth mentioning about the uploaded data, is that they are stored in a running MongoDB instance. Whenever needed, the user could re-visualize the dataset using the "show dataset" functionality. Eventually the user could create several instances of the anonymizer and submit a different query to a different dataset.

We create an optimized mode TLAQ query as shown in Figure5.13:

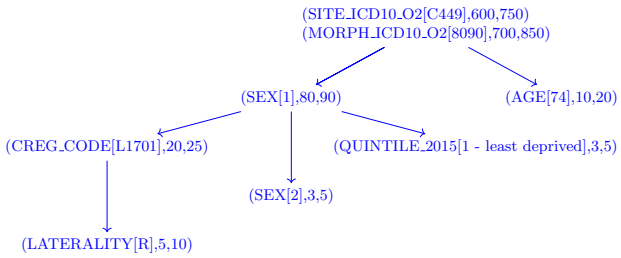


Fig. 5.13: Optimized mode TLAQ Example

And we execute it over a mignified version of Simulacrum's Tumour dataset to show how the use could be used. Following are figures related to the AB-DOM GUI interfaces showing different interfaces a user could leverage to enter input information such as the QID attributes, the sampling parameter or the TLAQ query (in JSON).

The Example TLAQ optimized mode query is depicted in Figure5.13. We can note that in the root node we have two constraints related to the attributes *SITE_ICD10_O2* and *MORPH_ICD10_O2* while the rest of the nodes specifies only one constraint which is inline with our definition of TLAQ optimized mode structure.

Using the AB-DOM graphical tool, we can create the TLAQ using the side bar tree component as shown in Figure 5.21 or we could upload the TLAQ in JSON format (e.g. as in Figure5.14) as shown in Figure 5.22. Furthermore it is possible to preview the structure of the created TLAQ using the Preview Query functionnality as shown in Figure 5.20.

```
{

    "outer": {
        "divaConstraints": [{
                "name": "RootNodeQ1",
                "constraintAttribute": "SITE_ICD10_O2",
                "constraintValue": "C449",
                "minBound": "600",
                "maxBound": "750"
            },
            {
                "name": "RootNodeQ1",
                "constraintAttribute": "MORPH_ICD10_O2",
                "constraintValue": "8090",
                "minBound": "700",
                "maxBound": "850"
            }
        ],
        "children": [{
        "name": "ChildNode1Q1",
        "constraintAttribute": "SEX",
        "constraintValue": "1",
        "minBound": "80",
        "maxBound": "90",
        "children": [{
            "name": "ChildNode12Q1",
            "constraintAttribute": "CREG_CODE",
            "constraintValue": "L1701",
            "minBound": "20",
            "maxBound": "25",
            "children": [{
                "name": "ChildNode13Q1",
                "constraintAttribute": "LATERALITY",
                "constraintValue": "R",
                "minBound": "5",
                "maxBound": "10",
                "children": []
        }]
    },{
        "name": "ChildNode21Q1",
            "constraintAttribute": "SEX",
            "constraintValue": "2",
            "minBound": "3",
            "maxBound": "5",
            "children": []
    },{
        "name": "ChildNode22Q1",
            "constraintAttribute": "QUINTILE_2015",
            "constraintValue": "1 - least deprived",
            "minBound": "3",
            "maxBound": "5",
            "children": []
    }]
    },{
        "name": "ChildNode23Q1",
        "constraintAttribute": "AGE",
        "constraintValue": "74",
        "minBound": "10",
        "maxBound": "20",
        "children": []
    }]
    }


}
```


Fig. 5.14: Example of TLAQ query in JSON

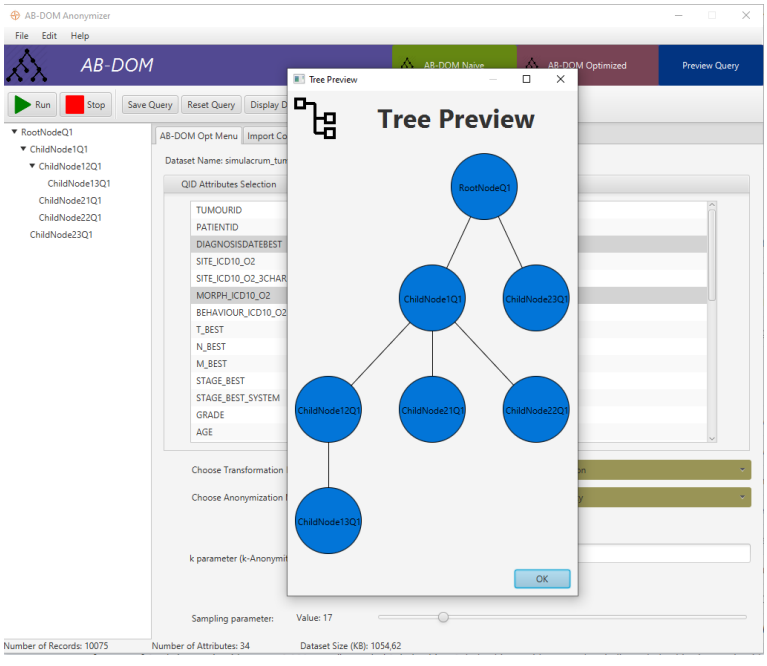


Fig. 5.20: Preview Query

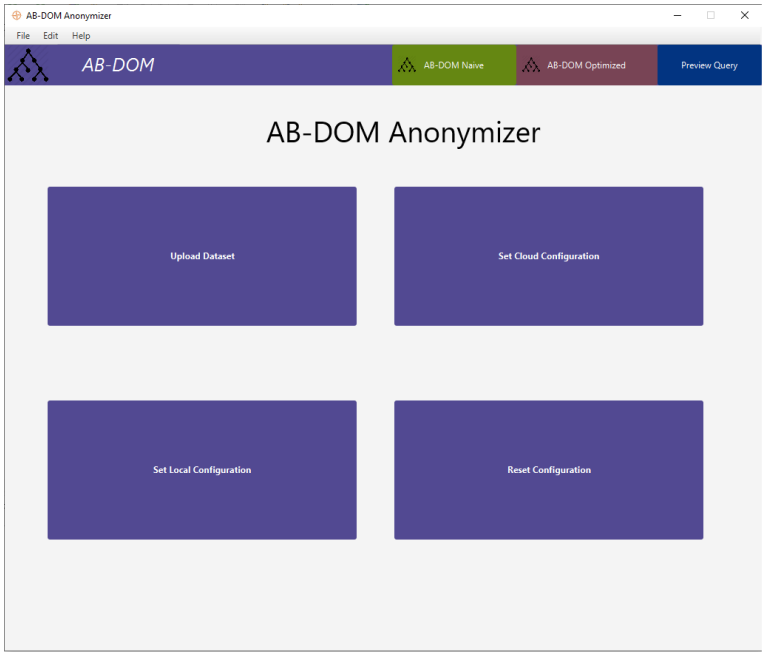


Fig. 5.15: Main User-interface

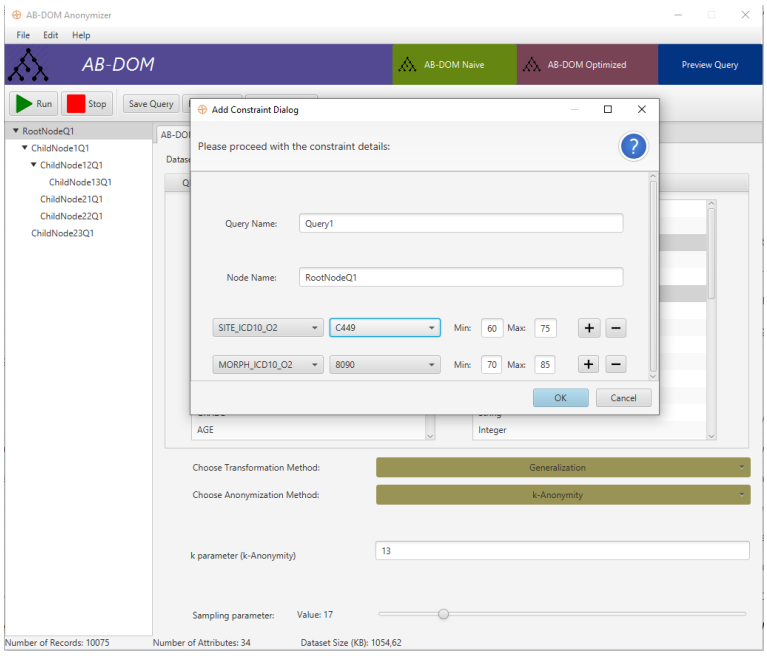


Fig. 5.21: Specifying Root Node Constraints

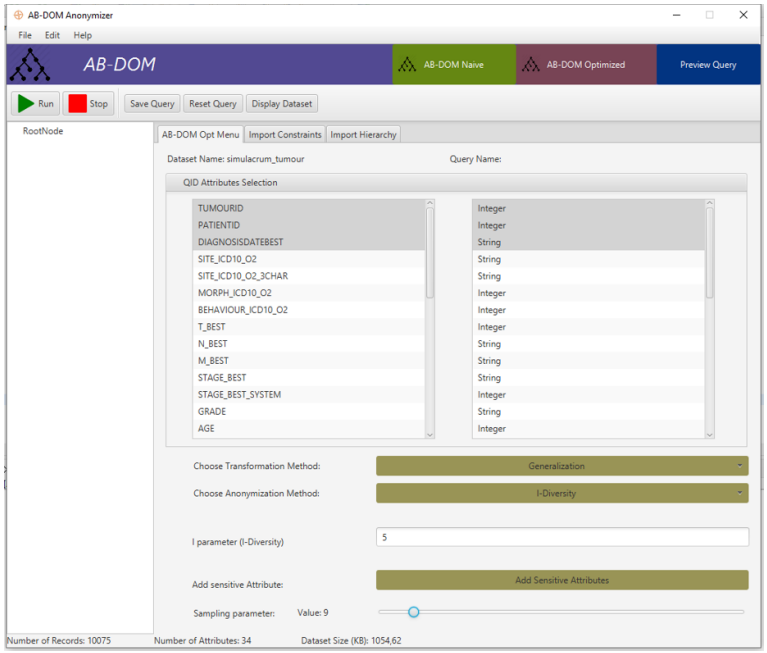


Fig. 5.16: Selecting the QID attributes

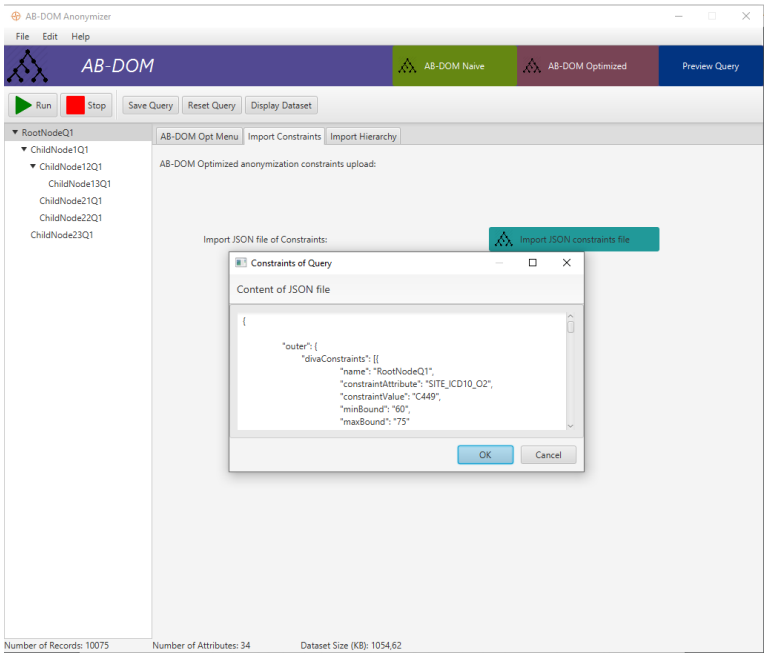


Fig. 5.22: Upload TLAQ through JSON

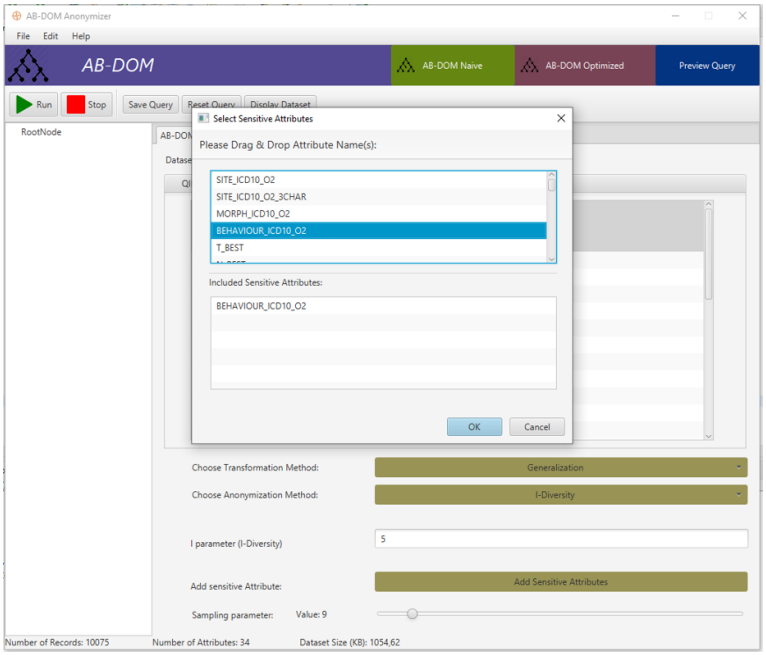


Fig. 5.17: Selecting the Sensitive Attributes

5.4.1 Anonymization Hierarchies

Anonymization techniques such as k-anonymity or l-diversity support the generalization of attribute values as a way to achieve anonymity (besides suppression). To enable this feature in the AB-DOM tool, we implement a dedicated interface where the user can upload a hierarchy for the attribute values of the considered dataset. Such hierarchy would consider the attribute values to anonymize (old values) and the target values (generalized new values). For that purpose, the interface depicted in Figure 5.23 shows the hierarchy uploaded in the GUI ready to be used by the underlying processing function.

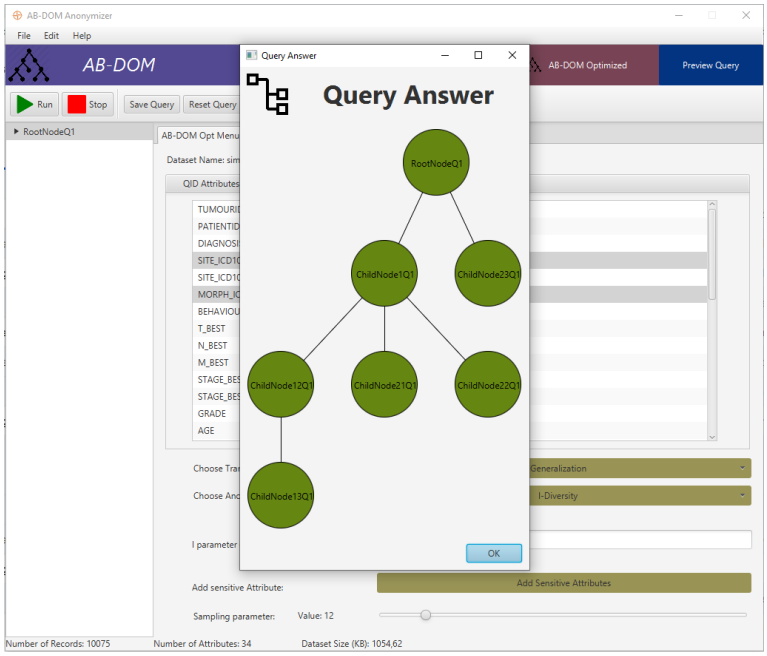


Fig. 5.18: AB-DOM Query Answer User Interface

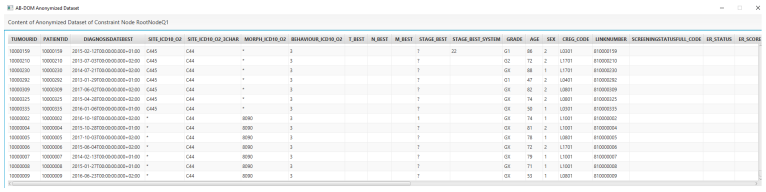


Fig. 5.19: Root Node Anonymized Dataset

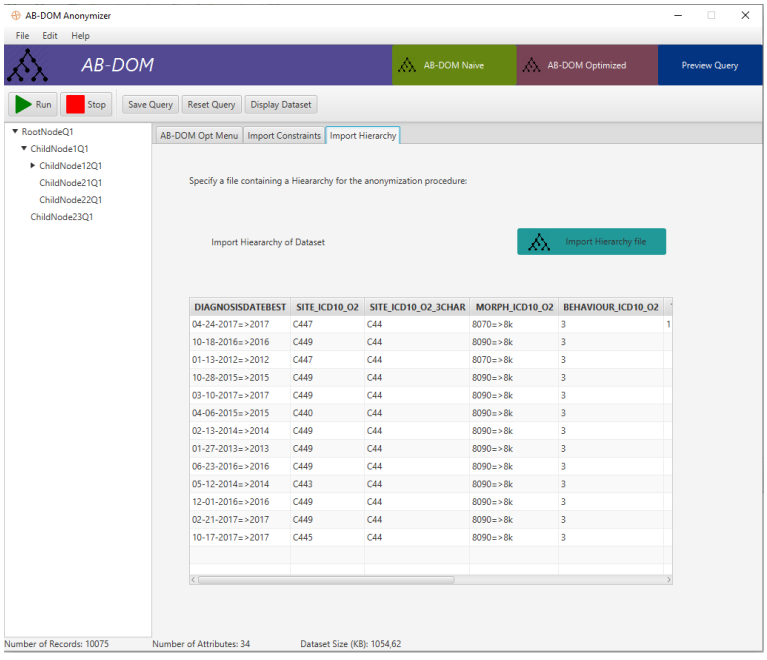


Fig. 5.23: Hiearchy of Attribute Values

In algorithm parlance, the anonymization works as follows: if the attribute value has a corresponding generalizing value as stated by the hierarchy then it will replace it with the corresponding value (considering the hiearchy as a Map string) where each iteration over a QID attribute (*qidColName*) and given a hierarchical transformation for an attribute value would be, in Spark code, like follows:

```
dataset = dataset
.withColumn("asMap",
    functions.expr("str_to_map(hierarchicalTransformationForOneValue,',',';')"))
.withColumn(qidColName,functions.expr("ifnull(asMap[" + qidColName +
    "]," + qidColName + ")"));
```

The first column transformation ensures that the column to be used for transforming the value in question would be in the right format (through a Map) and the second column transformation would ensure the actual replacement of the attribute value with their generalized form (as dicatated by the hierarchy) only if a transformation is defined for that particular attribute value.

Otherwise the anonymization falls back to the standard suppression way which, in Spark ,would translate into the following logic (applies for categorical attributes):

```
dataset = dataset.withColumn(columnName, functions.expr("concat_ws('',
    substring("+columnName+", 0,
    floor(char_length("+columnName+")/2)), '*')"));
```

This will hide half of the attribute value while leaving the other half as it is. Full suppression would simply replace the attribute value with a '*'.

After specifying all input parameters mmentionend, we could run the AB-DOM processing to eventually obtain the results in tree-shaped structured as shown in Figure 5.18. To donwload the resulting anonymized and diversity preserved datasets we could simply select the target nodes. Example of such resulting datasets (for root node) is shown in Figure 5.19.

5.5 Experimental Assessment and Analysis

In this Section, we provide our comprehensive experimental evaluation and analysis of AB-DOM algorithm, against both synthetic and real-life datasets, all in the healthcare domain. The derived experimental results allow us to test the effectiveness of our algorithm which is demanded to run within the internal layer of the QUALITOP big data lake.

5.5.1 Experimental Setup

AB-DOM has been implemented on top of the Cloud machine of the *Big Data Engineering and Analytics Laboratory* (iDEA Lab) of the University of Cal-

abria, Rende, Italy, called *iDEACloud*. iDEACloud is composed by 21 virtual machines (VM), each one equipped with *8-Core Intel Xeon Gold 5120* CPU @ 2.2 GHz and 32GB RAM. The running operating system is *Microsoft Windows Server 2019*. On top of this VM *cluster*, the Cloud environment is based on *Apache Hadoop* (version 3.2.2) [3]. The big data processing framework *Apache Spark* (version 3.1.2) [4] completes our Cloud software infrastructure, on top of which AB-DOM has been developed. Particularly, in our implementation, we adopted *Apache YARN* (version 3.2.2) [4] as Cloud resource manager. Finally, the iDEACloud is accessed remotely via commodity client machines running *Microsoft Windows 10* as operating system. Figure 5.24 shows an excerpt of the experimental architecture we built up, where 3 cloud nodes of iDEACloud are represented.

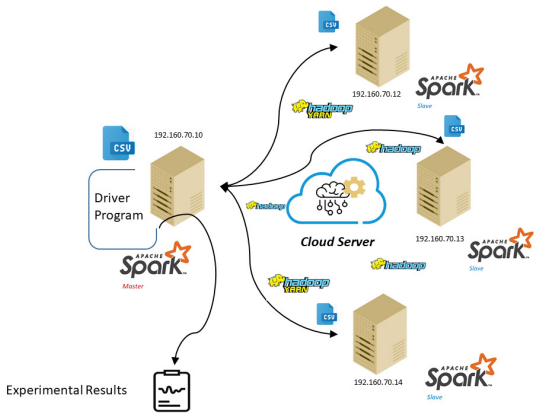


Fig. 5.24: An Excerpt of the Experimental Architecture

5.5.2 Implementation Details

AB-DOM has been implemented based on *Java* (version 8) in the *Eclipse* (version 4.21.0) software development environment. AB-DOM's implementation adheres to the *MapReduce big data processing paradigm* [74]. According to this paradigm, input big healthcare datasets are partitioned across the target iDEACloud nodes, namely *datanodes* in the Hadoop's vocabulary. The algorithm is launched from a client machine that connects to the master iDEACloud node via YARN, namely *namenode* in the Hadoop's vocabulary. Big data management, processing and mining libraries are executed via ad-hoc Spark libraries, which, at the end, realize the effective MapReduce paradigm. On the basis of this paradigm, "pieces" of computational tasks are executed on each datanode and partial results are combined at the namenode.

5.5.3 Synthetic Healthcare Datasets

In our experimental campaign, we first considered the following three synthetic datasets in the healthcare domain:

- $SBHD_U$;
- $SBHD_G$;
- $SBHD_Z$;

such that: (*i*) $SBHD_U$ stores healthcare data distributed according to a *Uniform distribution* [47]; (*ii*) $SBHD_G$ stores healthcare data distributed according to a *Gauss distribution* [47]; (*iii*) $SBHD_Z$ stores healthcare data distributed according to a *Zipf distribution* [47].

In the following, we provide a detailed description of these datasets.

In order to generate our synthetic datasets, we applied simple yet effective *sampling-based techniques* (e.g., [184]) over (small) real-life healthcare datasets, and, from these data, we generated our synthetic data. This allows us to capture the complexity of the investigated domain, by simply varying the *sampling generative function* $\Gamma(f)$ according to the three different distributions [47]: *Uniform*, denoted by $\Gamma(U)$, *Gauss*, denoted by $\Gamma(G)$, and *Zipf*, denoted by $\Gamma(Z)$.

In more details, let $A_i \in R$ be an attribute belonging to the schema of the current synthetic dataset R, we randomly populated the domain of A_i, denoted by $Dom(A_i)$, by sampling data from a small healthcare dataset, denoted by Y_{real}, according to the current sampling generative function $\Gamma(f)$, being $f \in \{U, G, Z\}$. Therefore, new sampled values for $Dom(A_i)$ are generated in the new sampled data domain, denoted by $Dom_{Sample}(A_i)$, and the new data domain of A_i, denoted by $Dom_{Gen}(A_i)$, is finally obtained as follows:

$$Dom_{Gen}(A_i) = Dom(A_i) \cup Dom_{Sample}(A_i) \tag{5.4}$$

such that:

$$Dom_{Sample}(A_i) = sample_{\Gamma(f)}(Y_{real}) \tag{5.5}$$

By iterating the above-described procedure for every attribute $A_i \in R$, then the final big synthetic dataset is obtained.

More into details, for the case of $\Gamma(U)$, we set two parameters, namely the lower bound L_1 and the upper bound L_2, such that $L_1 < L_2$, and we sampled data from Y_{real} via a Uniform distribution with parameter u uniformly varying over the interval $[L_1 : L_2]$. For the case of $\Gamma(G)$, we set four parameters, namely the lower bound μ_1 and the upper bound μ_2, such that $\mu_1 < \mu_2$, and the lower bound σ_1^2 and the upper bound σ_2^2, such that $\sigma_1^2 < \sigma_2^2$, respectively, and we sampled data from Y_{real} via a Gauss distribution with *mean* μ and *variance* σ^2, respectively, such that μ uniformly varies over the interval $[\mu_1 : \mu_2]$ and σ^2 uniformly varies over the interval $[\sigma_1^2 : \sigma_2^2]$. Finally, for the case of $\Gamma(Z)$, we set two parameters, namely the lower bound Z_1 and the upper

bound Z_2, such that $Z_1 < Z_2$, and we sampled data from Y_{real} via a Zipf distribution with parameter z uniformly varying over the interval $[Z_1 : Z_2]$.

Table 5.3 reports the cardinalities for all the synthetic healthcare datasets considered in our experimental campaign. As shown in Table 5.3, for each synthetic dataset, we achieved a number of tuples that we retained adequate for the scope of big data processing in Cloud computing environments (e.g., [198]). This also allows us to generate datasets that are, finally, totally convergent to the effective goal of our research.

Table 5.3: Cardinalities of the Synthetic Healthcare Datasets of the Experimental Campaign

Dataset	**Cardinality**
$SBHD_U$	1,200,000
$SBHD_G$	1,200,000
$SBHD_Z$	1,200,000

5.5.4 Real-Life Healthcare Datasets

In our experimental campaign, we considered the following three real-life datasets in the healthcare domain:

- *Immunotherapy* [5];
- *SEER Breast Cancer* [191];
- *Simulacrum* [166].

In the following, we provide a detailed description of these datasets.

Immunotherapy [5]. *Immunotherapy* stores information about wart treatment results of 90 patients using immunotherapy. Data were collected from patients with plantar and common warts, who have been referred to the dermatology clinic. These two types of warts are two of the most common wart types. The dataset has eight features collected when the immunotherapy method was employed.

SEER Breast Cancer [191]. *SEER Breast Cancer* provides information on population-based breast cancer statistics from the *Surveillance, Epidemiology, and End Results* (SEER) program of the *National Cancer Institute* (NCI) [191]. The dataset stores data about female patients with infiltrating duct and lobular carcinoma breast cancer diagnosed during 2006-2010. Some patients like those whose survival months were less than 1 month were excluded. Finally, 4024 patients have been selected in the dataset.

Simulacrum [166]. The *Simulacrum* is a *synthetic* cancer dataset that imitates data held securely within the *National Cancer Registration and Analysis Service* (NCRAS) of United Kingdom. Due to this engineering, *Simulacrum*

looks and feels like the real cancer data held within NCRAS but it does not contain any real patient information, for privacy constraints. However, although data are synthetic, *Simulacrum* maintains most of the properties of the original data, with a high degree of accuracy. More specifically, it contains data about patients diagnosed over a two-year period, such as age and sex, and data about tumors, such as staging and pathology information. Therefore, *Simulacrum* can be considered very "close" to a real-life dataset. In our experimental campaign, we focused the attention on the main table of *Simulacrum*, namely *Tumor*, which contains detailed tumor data for a total of $2.4M$ records. More into details, staging and pathology of tumors are represented, with also the possibility of a patient having more than one tumor.

5.5.5 Cloud-Enabled Real-Life Big Healthcare Datasets

In order to make our real-life datasets *Immunotherapy* and *SEER Breast Cancer* suitable to a reliable experimental assessment and evaluation in Cloud environments, and to expected big datasets' characteristics, like done for the case of synthetic datasets (see Section 5.5.3), we significantly extended the cardinalities of these datasets by again applying sampling-based techniques. For what regards the real-life dataset *Simulacrum*, the original number of records (i.e., $2.4M$) has been considered suitable for the goals of the experimental analysis.

Basically, in a similar manner to the case of synthetic datasets, given a dataset R, an attribute A_i belonging to the schema of R, and the data domain of A_i, denoted as $Dom(A_i)$, we simply generate new values from $Dom(A_i)$ via uniformly sampling $Dom(A_i)$ itself, thus achieving a new sampled data domain, denoted by $Dom_{Sample}(A_i)$. The new data domain of A_i, denoted by $Dom_{Gen}(A_i)$, is finally generated as follows:

$$Dom_{Gen}(A_i) = Dom(A_i) \cup Dom_{Sample}(A_i) \tag{5.6}$$

It should be noted that, since we use the same original data domain $Dom(A_i)$ to generate the sampled data domain $Dom_{Sample}(A_i)$, the "quality" of the final data domain $Dom_{Gen}(A_i)$ is completely coherent with the intrinsic characteristics of a real-life data domain. Obviously, if a *set of constraints* exists over A_i, denoted as $C(A_i)$, as applied to the data domain $Dom(A_i)$, then the artificial data generation procedure ensures that $C(A_i)$ is also verified on A_i as applied to $Dom_{Gen}(A_i)$.

By iterating the above-described procedure to every attribute A_i of R, then the final Cloud-enabled big dataset R_{Cloud} is obtained. In our experimental campaign, the following Cloud-enabled real-life datasets have been generated from the target real-life datasets *Immunotherapy* and *SEER Breast Cancer*, named as $Immunotherapy_{Cloud}$ and $SEERBreastCancer_{Cloud}$, respectively. As in a uniform vision, Table 5.4 reports the cardinalities for all the real-life healthcare datasets considered in our experimental campaign.

Table 5.4: Cardinalities of the Real-Life Healthcare Datasets of the Experimental Campaign

Dataset	**Cardinality**
$Immunotherapy_{Cloud}$	300,000
$SEERBreastCancer_{Cloud}$	1,000,000
$Simulacrum$	2,400,000

5.5.6 Metrics and Experimental Parameters

In our experimental campaign, we considered several experimental metrics that aim at precisely assessing and evaluating the various aspects of our AB-DOM algorithm. In particular, among these, the most important ones are: (i) supporting diversity while anonymizing data (as the basis of the underlying precision medicine process); (ii) ensuring performance and scalability of main algorithms while dealing with big healthcare datasets. Therefore, the experimental assessment and analysis of AB-DOM has focused on these two main experimental metrics.

In particular, as regards the first metrics, inspired by [28], we introduced the following *accuracy metrics* that measures the level of anonymization of a target data-anonymization technique T by comparing the original dataset R and the anonymized dataset R' produced by T. Of course, in our case, the diversity constraints apply to the overall anonymization process, so that they are "included" in the measure of the accuracy degree of the (data) anonymization process.

More into details, given a dataset R and its anonymized counterpart R', the anonymization accuracy between R and R', denoted by $AA(R, R')$, is defined as follows:

$$AA(R', R) = \frac{1}{\sum_{i=0}^{|R|} |I_{t_i,R'_{\neg *}}|^2 + |R| \sum_{i=0}^{|R|} |I_{t_i,R'_*}|} \tag{5.7}$$

such that: (i) $R'_* \subseteq R$ is the set of all suppressed tuples in R; (ii) $R'_{\neg *} \subseteq R$ is the set of all non-suppressed tuples in R; (iii) $|R|$ is the number of tuples in R; (iv) $|R'|$ is the number of tuples in R'; (v) $|R'_*|$ is the number of suppressed tuples in R' (see Section 5.1); (vi) $|R'_{\neg *}|$ is the number of non-suppressed tuples in R'; (vii) t_i: i-th tuple in R; $(viii)$ t'_i: i-th tuple in R'; (ix) $t'_{*,i}$ is the i-th suppressed tuple in R'; (x) $t'_{\neg *,i}$ is the i-th non-suppressed tuple in R'; (xi) I_{t_i,R'_*} is the set of tuples in R'_* that are indistinguishable from t_i in R; (xii) $I_{t_i,R'_{\neg *}}$ is the set of tuples in $R'_{\neg *}$ that are indistinguishable from t_i in R; $(xiii)$ $|I_{t_i,R'_*}|$ is number of tuples in I_{t_i,R'_*} ; (xiv) $|I_{t_i,R'_{\neg *}}|$ is the number of tuples in $I_{t_i,R'_{\neg *}}$. Particularly, since, as described in Section 5.3, the output of AB-DOM is represented by a collection of diversity-aware anonymized datasets indexed by a suitable tree-like indexing data structure, we simply considered

as final metrics the *average* anonymization accuracy computed on top of all the output datasets.

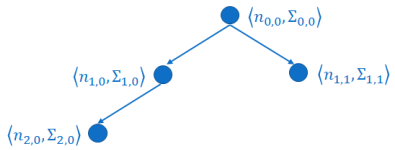


Fig. 5.25: Structure of the Synthetic TLAQ used in the Experimental Campaign

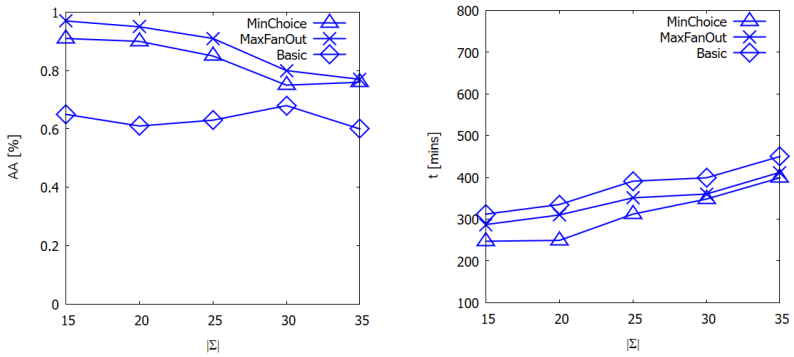
Now, we focus the attention on the other experimental parameters of our experimental campaign. First, from previous formal modeling provided throughout the paper, (*i*) R models the target big healthcare dataset; (*ii*) $|R|$ models the cardinality of R; (*iii*) Σ models the set of diversity constraints; (*iv*) $|\Sigma|$ models the cardinality of Σ; (*v*) k models the fundamental parameter of the baseline k-anonymity procedure [188]. In addition, similarly to the experimental analysis in the DIVA proposal [138], we introduce the conflict rate (*cr*) concept. *cr* measures the number of overlapping relevant tuples between a pair of diversity constraints, as first, and, then, the definition is extended to the entire set of diversity constraints Σ by applying to pairs of constraints. *cr* ranges over [0:1], where 0 indicates no overlap and 1 full overlap.

5.5.7 Experimental Results

In our experimental campaign, for every of (synthetic and real-life) big dataset in the healthcare domain, namely $SBHD_U$, $SBHD_G$, $SBHD_Z$, $Immunotherapy_{Cloud}$, $SEERBreastCancer_{Cloud}$ and $Simulacrum$, we performed three kinds of experiments, by combining metrics and experimental parameters described in the previous Sections. For each kind of experiment, when we selected a metrics m as output and an experimental parameter p as ranging input parameter, we fixed all the other experimental parameters to a certain value. In the first kind of experiment, we measure the anonymization accuracy AA with respect to the ranging of the number of diversity constraints $|\Sigma|$. It is almost obvious that, by increasing the number of diversity constraints, the anonymization accuracy is stressed more. In the second kind of experiment, we measure the time t necessary to evaluate the collection of synthetic TLAQ of structure shown in Figure 5.25, still with respect to the number of diversity constraints $|\Sigma|$. In this case, the more is the number of diversity constraints, the more is the time needed to evaluate input TLAQ against the target big healthcare dataset, due to the intrinsic modus operandi of the core DIVA algorithm encoded within AB-DOM (see Section 5.1 and Section 5.3). Finally, in the third kind of experiment, we measure again the

anonymization accuracy AA with respect to the ranging of the conflict rate cr, which is modeled as a percentage with respect to the total number of diversity constraints. It should be noted that this latter analysis aims at testing the *robustness* of AB-DOM with respect to the applicative case of considering more "probing" input datasets (i.e., datasets for which the diversity constraints are more and more difficult to be satisfied).

First, we consider the case of synthetic big healthcare datasets. Figure 5.26(*(a)*) shows the variation of the anonymization accuracy AA with respect to the ranging of the number of diversity constraints $|\Sigma|$, for all the three variants of the core DIVA algorithm (i.e., *MinChoice*, *MaxFanOut*, *Basic* – see Section 5.1), when the synthetic dataset $SBHD_U$ is considered. Figure 5.26(*(b)*) shows instead the variation of the time t with respect to the ranging of the number of diversity constraints $|\Sigma|$, still for the dataset $SBHD_U$ and for all the three variants of the core DIVA algorithm. Finally, Figure 5.27(*(a)*) shows the variation of the anonymization accuracy AA with respect to the ranging of the conflict rate cr, again for the dataset $SBHD_U$ and for all the three variants of the core DIVA algorithm.



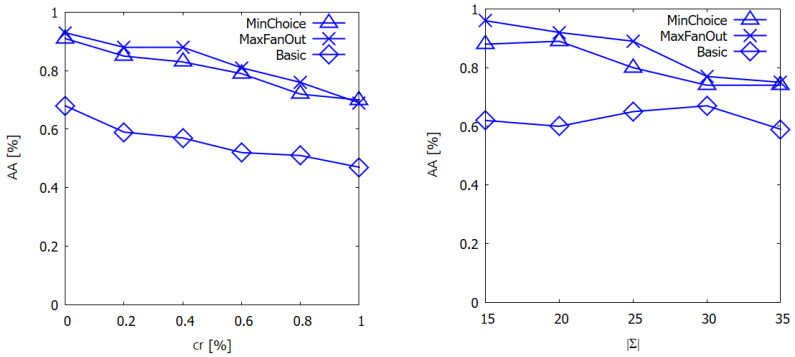
(a) (b)

Fig. 5.26: Anonymization Accuracy AA vs Number of Diversity Constraints $|\Sigma|$ for the $SBHD_U$ Dataset (a) - Time t vs Number of Diversity Constraints $|\Sigma|$ for the $SBHD_U$ Dataset (b)

Figure 5.27(*(b)*), Figure 5.28(*(a)*) and Figure 5.28(*5.28b*) show the same experimental pattern for the $SBHD_G$ synthetic dataset, respectively.

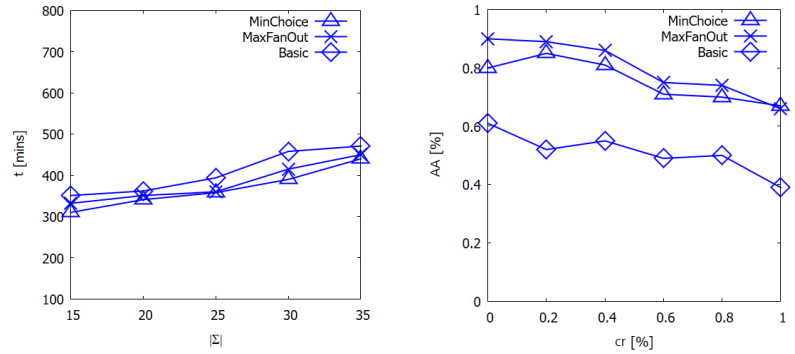
Finally, Figure 5.29(*(a)*), Figure 5.29(*(b)*) and Figure 5.30(*5.30a*) show the same experimental pattern for the $SBHD_Z$ synthetic dataset, respectively.

Now, we focus the attention on real-life big healthcare datasets. Similarly to the case of synthetic datasets, we show the three different experimental patterns we engineered in our experimental campaign. Figure 5.30(*(b)*) shows the variation of the metrics AA with respect to the ranging of the



(a) (b)

Fig. 5.27: Anonymization Accuracy AA vs Conflict Rate cr for the $SBHD_U$ Dataset (a) - Anonymization Accuracy AA vs Number of Diversity Constraints $|\Sigma|$ for the $SBHD_G$ Dataset (b)



(a) (b)

Fig. 5.28: Time t vs Number of Diversity Constraints $|\Sigma|$ for the $SBHD_G$ Dataset (a) - Anonymization Accuracy AA vs Conflict Rate cr for the $SBHD_G$ Dataset (b)

experimental parameter $|\Sigma|$, for all the three variants of the core DIVA algorithm (i.e., MinChoice, MaxFanOut, Basic – see Section 5.1), when the dataset $Immunotherapy_{Cloud}$ is considered. Figure 5.31(*(a)*) shows instead the variation of the metrics t with respect to the experimental parameter $|\Sigma|$, still for the dataset $Immunotherapy_{Cloud}$ and for all the three variants of the core DIVA algorithm. Finally, Figure 5.31(*(b)*) shows the variation of the metrics AA with respect to the ranging of the experimental parameter cr, again for

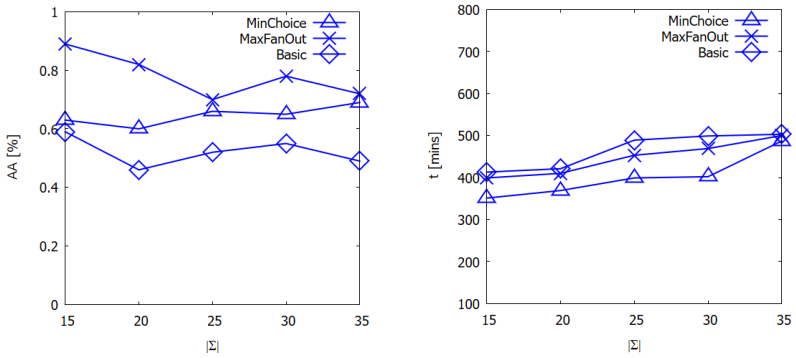


Fig. 5.29: Anonymization Accuracy AA vs Number of Diversity Constraints $|\Sigma|$ for the $SBHD_Z$ Dataset (a) - Time t vs Number of Diversity Constraints $|\Sigma|$ for the $SBHD_Z$ Dataset (b)

the dataset $Immunotherapy_{Cloud}$ and for all the three variants of the core DIVA algorithm.

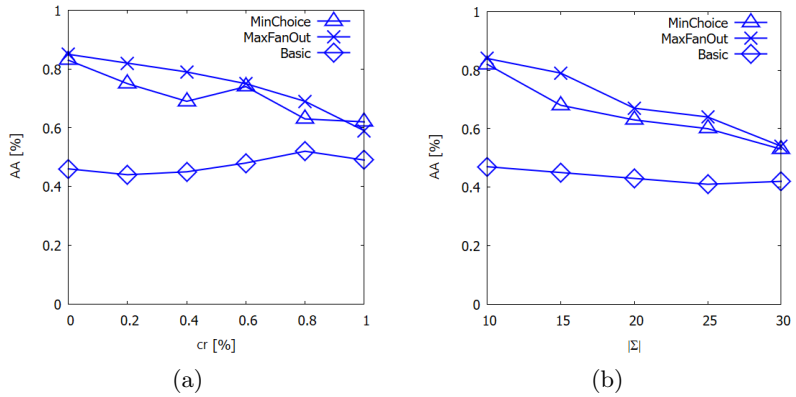
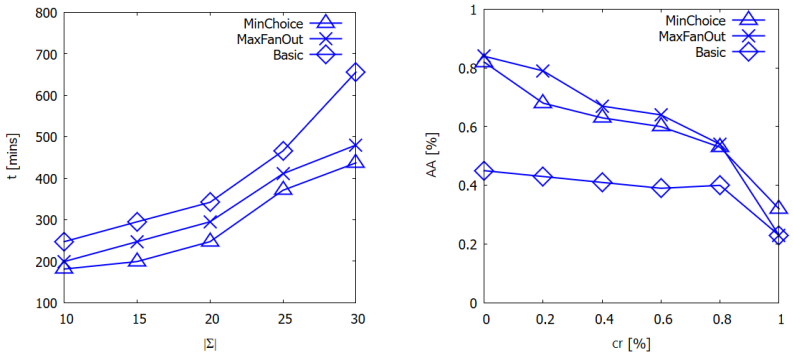


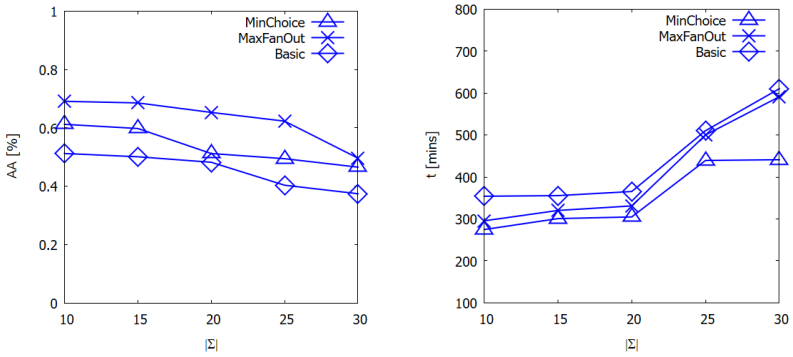
Fig. 5.30: Anonymization Accuracy AA vs Conflict Rate cr for the $SBHD_Z$ Dataset (a) - Anonymization Accuracy AA vs Number of Diversity Constraints $|\Sigma|$ for the $Immunotherapy_{Cloud}$ Dataset (b)

Figure 5.32(*(a)*), Figure 5.32(*(b)*) and Figure 5.33(*(a)*) show the same experimental pattern for the $SEERBreastCancer_{Cloud}$ dataset, respectively.



(a) (b)

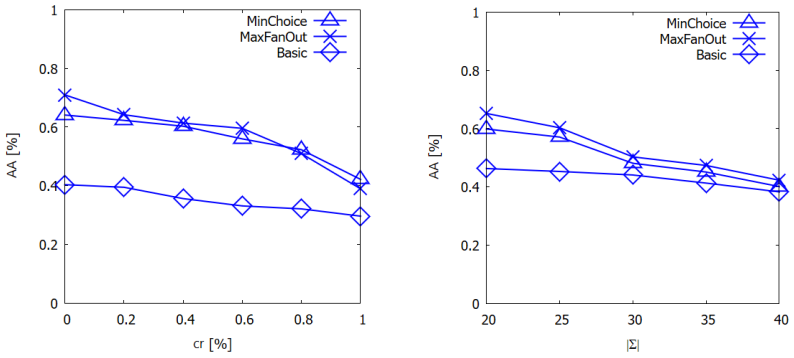
Fig. 5.31: Time t vs Number of Diversity Constraints $|\Sigma|$ for the $Immunotherapy_{Cloud}$ Dataset (a) - Anonymization Accuracy AA vs Conflict Rate cr for the $Immunotherapy_{Cloud}$ Dataset (b)



(a) (b)

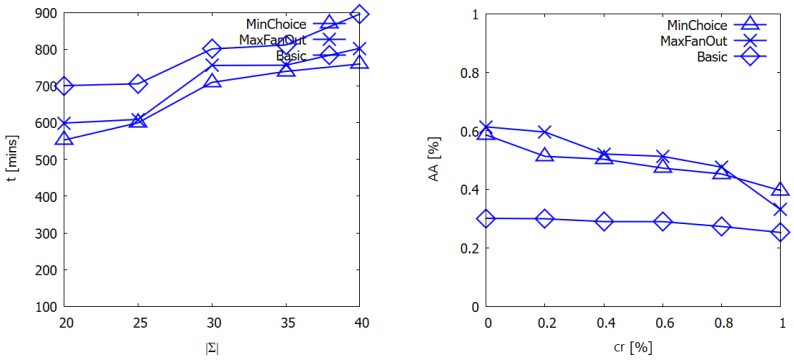
Fig. 5.32: Anonymization Accuracy AA vs Number of Diversity Constraints $|\Sigma|$ for the $SEERBreastCancer_{Cloud}$ Dataset (a) - Time t vs Number of Diversity Constraints $|\Sigma|$ for the $SEERBreastCancer_{Cloud}$ Dataset (b)

Finally, Figure 5.33(*(b)*), Figure 5.34(*(a)*) and Figure 5.34(*(b)*) show the same experimental pattern for the *Simulacrum* dataset, respectively.



(a) (b)

Fig. 5.33: Anonymization Accuracy AA vs Conflict Rate cr for the $SEERBreastCancer_{Cloud}$ Dataset (a) - Anonymization Accuracy AA vs Number of Diversity Constraints $|\Sigma|$ for the $Simulacrum$ Dataset (b)



(a) (b)

Fig. 5.34: Time t vs Number of Diversity Constraints $|\Sigma|$ for the $Simulacrum$ Dataset (a) - Anonymization Accuracy AA vs Conflict Rate cr for the $Simulacrum$ Dataset (b)

5.6 Summary

AB-DOM was designed to deal with hierarchical healthcare data as the main type of data we use in our research. Specifically, two modes of execution were proposed, the first is a standard approach through the TLAQ querying technique, the second is resource optimizing processing of TLAQ. In fact, in the more probing second case, we employ a differential privacy sampling method to benefit from reducing the execution cost over the cloud and at the same

time to process the data in a private way thus dealing with resource optimization challenges for data anonymization goals. By exposing its functioning over the cloud, we also deal with scalability matters, as data to process could be of high amount. Data indexation in data lake environment could benefit from AB-DOM as TLAQ enable the indexation of data through its constraint based model tailored to enhance the outcome of data analytics processes. In the next chapter, we will deal with another type of data and define a privacy preserving technique for data analytics in the case of data warehousing.

6

Drill-CODA: A Framework for Supporting Drill-Across Multidimensional Big Data Analytics over Big Co-Occurrence Aggregate Hierarchical Data

As we move through this thesis, we have shown how data analytics can benefit from the privacy preserving process that is generally applied in a prior step to analytics in the case of hierarchical data. In this chapter, we will define and discuss our OLAP based privacy preserving data analytics technique named *Drill*-CODA, and demonstrate its effectiveness in deriving analytics while keeping the queried data unrevealed.

Nowadays, *big data analytics* [172] is gaining momentum as one of the most relevant innovations of the last years. Actually, a lot of proposals appeared in literature, each one based on a specific *data mining* or *machine learning* technique/algorithm. The final goal is that of extracting useful and actionable knowledge from large-scale, *3v* big data repositories by overcoming well-understood limitations of (traditional) SQL-shaped knowledge discovery methodologies. Indeed, these kind of repositories expose relevant challenges to deal with, among which *volume* is not the most probing one, being well-accompanied by *velocity*, *variety*, and *value*. In addition, Data warehouses are an integral part of data lakes according to many literature works such as in [212]. *ETL* processes could be applied on data residing in data lakes in order to derive insights in a multi-dimensional way. On the other hand, given the role of *multidimensional big data analytics* in helping comprehend observations in a wide variety of topics ranging from *healthcare* to *experimental apparatuses*, from *environmental science* to *finance*, and so forth, developing solutions to extract such information from the data would lead to improved and enhanced comprehension over the processes in those areas that would, eventually, lead to upgraded solutions that are more tailored to the problems to solve related to the observations. Thankfully, a relevant boost is given to these methodologies through the impressive growth of *Cloud Computing* technologies, especially when coupled with big data paradigms (e.g., [210]), which would eventually, enlarge the spectrum of proposed solutions and the obtained results.

Basically, multidimensional big data analytics predicates the engrafting of *multidimensional analysis* principles within the target big data analytics process, by taking advantages from a greater expressive power and a greater ac-

curacy during the decision phase. Indeed, the foundations of this methodology proposes the usage of *dimensions* and *measures* to model the target application domain, thus constructing a powerful *multidimensional space* that allows us to gain into efficacy during the big data analytics phase. Coupled with this, specific big multidimensional data management and analytics algorithms must be devised and developed, beyond the actual *scan-based model* of major big data processing approaches. This evidence has been widely demonstrated by many authoritative studies in the area.

Inspired by this main context, in this chapter we address a specific problem: *how to provide powerful big data analytics over input aggregate multidimensional hierarchical data, while still ensuring privacy-preservation?*. Before going into details, it should be considered that the latter data analytics setting is relevant for a plethora of application scenarios, ranging from *urban analytics* to *social network analysis*, from *bio-medical tools* to *industry 4.0 prognostic tools*, and so forth. This because data of real-life settings are hierarchical by nature.

More into details, given a set of *hierarchical* big datasets, denoted by $\mathcal{S} = \{S_0, S_1, \ldots, S_{|\mathcal{S}|-1}\}$, the final goal consists in supporting *multidimensional* big data analytics over data in $\mathcal{S}$, while also ensuring the preservation of privacy of data items in $S_j \in \mathcal{S}$, such that $k \in \{0, 1, \ldots, |\mathcal{S}| - 1\}$. More into details, we aim at providing *drill-across* multidimensional big data analytics, being drill-across one well-know, traditional *Business Intelligence* (BI) operator supporting powerful analytics across different yet related data domains. Therefore, from the marriage of these two metaphors, our overall research proposal conveys in what we name as *drill-across multidimensional big data analytics*.

More broadly, the twofold goal consisting in realizing both *powerful analytics and privacy preservation* is reached through the framework *Drill*-CODA, which defines a complex approach that supports drill-across multidimensional big data analytics over the target collection of hierarchical big datasets whose content is *anonymized* thanks to a special *co-occurrence analysis technique* (e.g., [103]). Thus, we finally obtain a complex framework able of *supporting privacy-preserving drill-across multidimensional big data analytics over big co-occurrence aggregate hierarchical data*: *Drill*-CODA.

Key Contributions

This chapter is a summary of the proposed framework *Drill-Coda* framework and its related experimental assessments, which allow us to *support drill-across multidimensional big data analytics over big co-occurrence aggregate hierarchical data, with privacy-preservation features*. *Drill*-CODA is a composite framework that combines several data processing and analytics metaphors over hierarchical data, all in the multidimensional fashion, with the goal of providing useful insights over large-scale big data repositories, while protecting their privacy. The latter is, as for now, a critical challenge in the big data

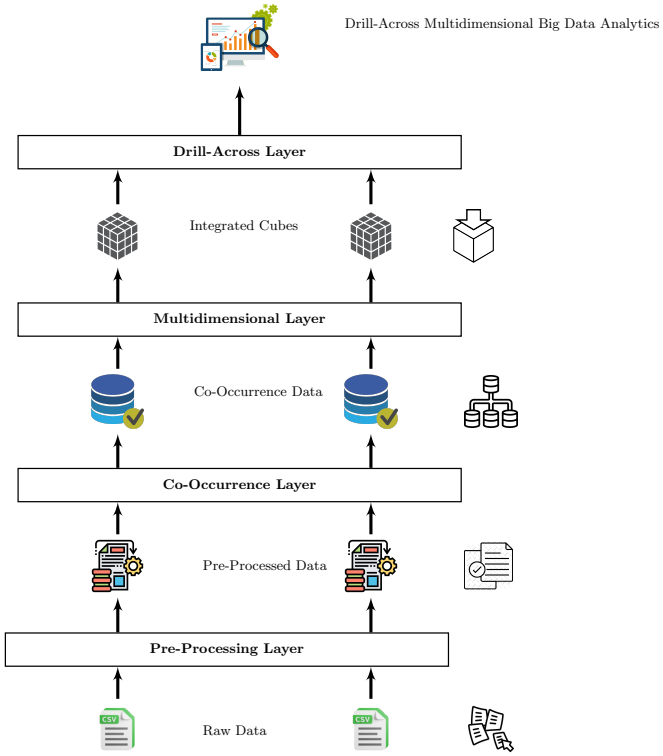


Fig. 6.1: The *Drill*-CODA Framework Data Processing Workflow

research community, which arises in a plethora of emerging big data application scenarios, ranging from *urban analytics* to *social network analysis*, from *bio-medical tools* to *industry 4.0 prognostic tools*, and so forth. This because data of real-life settings are hierarchical by nature. We complete our analytical contributions by means of a wide experimental analysis on real-life big datasets, which confirms the benefits coming from our proposed framework.

In a nutshell, the main components of the framework can be described as follows (see Figure 6.1):

- *pre-processing*: here, the input hierarchical big datasets are pre-processed in order to prepare them for the next steps, mostly exploiting multidimensional data processing methods;
- *co-occurrence analysis*: here, we apply co-occurrence analysis over the pre-processed datasets in order to shape data for the drill-across analytics and, at the same, achieve the privacy-preservation goal;

- *multidimensional aggregation*: in this step, data are now aggregated within *multidimensional OLAP data cubes* [96], which realize the reference storage layer for the subsequent drill-across querying step;
- *drill-across querying*: lastly, aggregate data cube cells are drilled-across along *all* the hierarchical dimensions of the OLAP data cubes in order to support the final privacy-preserving drill-across multidimensional big data analytics over big co-occurrence aggregate hierarchical data.

Specifically, the drill-across query layer originates a collection of (drill-across) answers retrieved from input OLAP data cubes that, eventually, populate the so-called *full-dimensional correlation set* $\mathcal{D}_{\mathcal{CO}}(\mathcal{S})$. $\mathcal{D}_{\mathcal{CO}}(\mathcal{S})$ stores collections of *correlated aggregates* retrieved from the execution of the reference set of drill-across queries along *all* the hierarchical dimensions defined on $\mathcal{S}$. Starting from $\mathcal{D}_{\mathcal{CO}}(\mathcal{S})$, insightful big data analytics in terms of meaningful *correlation indexes*, namely the *Pearson correlation coefficient* and the *Spearman correlation coefficient* [46], which both are powerful methods to measure the correlation that exists among (correlated) aggregates retrieved by the drill-across query layer.

Chapter Organization

The remaining part of this chapter is organized as follows. Section 6.1 defines the related notions and concepts in order to comprehend *Drill*-CODA and introduces the composing parts of the main algorithm. In Section 6.2, we highlight a cloud-based architecture for *Drill*-CODA. Finally, Section 6.3 focuses on the evaluation of the algorithm and depicts the correlation-based plots obtained from the data-warehouse based executions.

Published Works

[60] Alfredo Cuzzocrea and Selim Soufargi. "Drill-CODA: A Framework for Supporting Drill-Across Multidimensional Big Data Analytics over Big Co-Occurrence Aggregate Hierarchical Data". In: Accepted at DATA 2024.

6.1 The *Drill*-CODA Framework: Concepts, and Definitions

In this Section, we provide a formal description of the overall *Drill*-CODA framework proposal, by specifically illustrating its concepts and definitions. *Drill*-CODA comprises several steps (i.e., pre-processing, co-occurrence analysis, multidimensional aggregation, drill-across querying). Here, we formally describe in details these steps.

6.1.1 Pre-Processing

In the *Drill*-CODA pre-processing step, the input hierarchical big datasets in $\mathcal{S}$ are treated for preparation for the next steps of the whole technique. First, we focus the attention on the anatomy of these datasets. Being hierarchical in nature, given a dataset $S_j \in \mathcal{S}$, some attributes $\mathcal{W}(\mathcal{S}) = \{A_{k_0}, A_{k_1}, \ldots, A_{k_{|\mathcal{W}(\mathcal{S})|-1}}\} \in S_j$ play the role of *dimensions* while some other attributes $\mathcal{M}(\mathcal{S}) = \{A_{h_0}, A_{h_1}, \ldots, A_{h_{|\mathcal{M}(\mathcal{S})|-1}}\} \in S_j$, such that $k_u \neq h_l \forall u \wedge l$, play the role of *measures* related to those dimensions. Given a dimension $A_{k_u} \in \mathcal{W}(\mathcal{S})$, a *dimensional hierarchy* $\mathcal{H}(A_{k_u})$ is defined on top of it, as follows: $\mathcal{H}(A_{k_u}) = \{l_{A_{k_u},0}, l_{A_{k_u},1}, \ldots, l_{A_{k_u},|\mathcal{H}(A_{k_u})|-1}\}$, such that $l_{A_{k_u},q}$ models a *hierarchical level* of $\mathcal{H}(A_{k_u})$, with $q \in \{0, 1, \ldots, DEPTH(\mathcal{H}(A_{k_u}))-1\}$, where $DEPTH$ is a multidimensional operator that retrives the depth of the hierarchy $\mathcal{H}(A_{k_u})$. To give an example, the hierarchy $Store \leftarrow City \leftarrow Region \leftarrow Country$ of the dimension $Zone$ could be related to the measure $Sale$. A possible instance is the following one: $Computer\,Parts \leftarrow Florence \leftarrow Tuscany \leftarrow Italy$ related to 825.00$. However, as it will be clearer through the paper, while we keep in our model to respect the property of *autonomicity*, we do not process neither use the measures of datasets $S_j \in \mathcal{S}$ directly, since our framework is oriented to more advanced analytics.

In the pre-processing step, given a dataset $S_j \in \mathcal{S}$, we define: (i) a set of *target attributes* of interest for the analysis, namely $\mathcal{T}_{S_j} = \{T_{S_j,0}, T_{S_j,1}, \ldots, T_{S_j,|\mathcal{T}_{S_j}|-1}\}$, and the respective set of attribute values of interest for the analysis, namely $\mathcal{V}_{S_j} = \{V_{S_j,0}, V_{S_j,1}, \ldots, V_{S_j,|\mathcal{V}_{S_j}|-1}\}$, such $T_{S_j,k} = V_{S_j,k}, \forall k \in \{0, 1, \ldots, |\mathcal{T}_{S_j}| - 1 = |\mathcal{V}_{S_j}| - 1\}$; (ii) a specific aggregate operator selected in the set $AO = \{SUM, COUNT, MIN, MAX, AVG\}$, which applies on top of the target attributes in $\mathcal{T}_{S_j}$; (iii) a set of *functional attributes* with respect to which the target attributes are analyzed, namely $\mathcal{F}_{S_j} = \{F_{S_j,0}, F_{S_j,1}, \ldots, F_{S_j,|\mathcal{F}_{S_j}|-1}\}$, such that $T_{S_j,k} \neq F_{S_j,h}, \forall k \neq h$.

Based on these definitions, we project S_j by target attributes in $\mathcal{T}_{S_j}$, and then we filter the obtained projected dataset by means of values in $\mathcal{V}_{S_j}$. After that, we apply the given aggregate operator in AO and we aggregate data of target attributes along *all* the hierarchies of dimensions in $\mathcal{W}(S_j)$. Of course, we aggregate the functional attributes in $\mathcal{F}_{S_j}$ as well. Formally, we denote the pre-processed dataset derived from S_j as S_j^{PP}, and we construct the set $\mathcal{S}^{PP} = \{S_0^{PP}, S_1^{PP}, \ldots, S_{|\mathcal{S}^{PP}|-1}^{PP}\}$.

To give an example, consider the schema :
$\{Day, Month, Year, City, Country, Gender, COUNT(Skin\,Cancer),$
$COUNT(Lung\,Cancer)\}$. A possible instance is the following:
$\{13, 11, 2020, Milan, Lombardy, Italy, M, 32, 69\}$, which models the event that, on $November$ 13, 2020, in $Milan$, 32 male (M) patients died due to $Skin\,Cancer$ and 69 male (M) patients died due to $Lung\,Cancer$, respectively.

6.1.2 Co-Occurrence Analysis

In the *Drill*-CODA co-occurrence analysis step, the final goal is that of obtaining the privacy-preservation effect, since we apply a kind of *co-occurrence-based anonymization technique* that takes advantage from the multidimensional nature of target data. Before going into details, to become convinced about the approach, consider the following toy example. Let $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$ be two big healthcare datasets that store patient events about diseases, treatments, therapies and so forth, being the latter all *sensitive data* whose privacy should be preserved. Here, it is interesting and natural to analyze *correlations* that may exist among data $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$, in order, for instance, to discover *cross-therapies* performed by *different* hospitals over the *same* diseases, in order to ameliorate the effectiveness of combined therapies, perhaps obtained from the merging of therapies of different hospitals. In this case, let *Location* and *Time* be two *co-occurrence attributes*, both belonging to the schemes of $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$, respectively. Given a specific death event, for instance caused by cancer, it is possible to compute two different *co-occurrence datasets* from $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$, namely $\mathcal{CO}[D_{i,\mathcal{H}}, D_{j,\mathcal{H}}, Location]$ and $\mathcal{CO}[D_{i,\mathcal{H}}, D_{j,\mathcal{H}}, Time]$, respectively, such that $\mathcal{CO}[D_{i,\mathcal{H}}, D_{j,\mathcal{H}}, Location]$ stores the death events of $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$ that refer to the *same Location*, while $\mathcal{CO}[D_{i,\mathcal{H}}, D_{j,\mathcal{H}}, Time]$ stores the death events of $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$ that refer to the *same* $Time$, respectively. It should be noted that both the two co-occurrence attributes *Location* and $Time$ model specific hierarchical levels of certain hierarchies associate to dimensions in both $D_{i,\mathcal{H}}$ and $D_{j,\mathcal{H}}$, respectively. Moreover, the co-occurrence analysis provides us with the desiderata privacy-preservation effect due to the fact that, when abstracted to the $Time$ level, e.g. $Year$, and the *Location* level, e.g. *Country*, individual data are anonymized while aggregate data still suffice to the big data analytics purposes.

Formally, given the set of pre-processed hierarchical big datasets $\mathcal{S}^{PP} = \{S_0^{PP}, S_1^{PP}, \ldots, S_{|\mathcal{S}^{PP}|-1}^{PP}\}$ and a set of common co-occurrence attributes $\mathcal{A}_{\mathcal{S},CO} = \{A_{\mathcal{S},CO,0}, A_{\mathcal{S},CO,1}, \ldots, A_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}\} \in S_j \in \mathcal{S}$, such that $A_{\mathcal{S},CO,k} \in S_j^{PP}, \forall S_j^{PP} \in \mathcal{S}^{PP}, \forall k \in \{0, 1, \ldots, |\mathcal{A}_{\mathcal{S},CO}| - 1\}$, we generate $|\mathcal{A}_{\mathcal{S},CO}|-1$ co-occurrence datasets, namely $\mathcal{CO}_{\mathcal{S},CO} = \{C_{\mathcal{S},CO,0}, C_{\mathcal{S},CO,1}, \ldots, C_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}\}$, such that each dataset $C_{\mathcal{S},CO,k} \in \mathcal{CO}_{\mathcal{S},CO}$ is defined as follows:

$$C_{\mathcal{S},CO,k} = \{A_{\mathcal{S},CO,k}, \langle F_{S_j,h}, \{AO_0(T_{S_j,0}), AO_1(T_{S_j,1}), \ldots, AO_{|\mathcal{T}_{S_j}|-1}(T_{S_j,|\mathcal{T}_{S_j}|-1})\}\rangle\}, \forall k \in \{0, 1, \ldots, |\mathcal{A}_{\mathcal{S},CO}| - 1\} \tag{6.1}$$

such that: (i) $A_{\mathcal{S},CO,k}$, where $k \in \{0, 1, \ldots, |\mathcal{A}_{\mathcal{S},CO}| - 1\}$ denotes a co-occurrence attribute; (ii) $F_{S_j,h}$, where $h \in \{0, 1, \ldots, |\mathcal{F}_{S_j}|-1\}$ denotes a functional attribute (see Section 6.1.1); (iii) AO_z, where $z \in \{0, 1, \ldots, |AO| - 1\}$, denotes an aggregate operator selected from the set AO (see Section 6.1.1).

To give an example, consider the schema of the first co-occurrence dataset, defined as follows: $\{Year, \langle Gender, COUNT(Skin\,Cancer), COUNT(Lung\,Cancer), COUNT(Diabetes\,Type\,1), COUNT(Diabetes\,Type\,2)\rangle\}$. A possible instance

is the following one: $\{2022, \{\langle F\text{-}Cancer, 35, 74\rangle, \langle M\text{-}Cancer, 37, 58\rangle, \langle M\text{-}Diabetes, 27, 51\rangle, \langle F\text{-}Diabetes, 43, 68\rangle\}$, which models the event that, during 2022, with *no* reference to the location, (i) a total of 109 female (F) patients died by cancer, specifically 35 of $Skin\,Cancer$ and 74 of $Lung\,Cancer$; (ii) a total of 95 male (M) patients died by cancer, specifically 37 of $Skin\,Cancer$ and 58 of $Lung\,Cancer$; (iii) a total of 78 male (M) patients died by diabetes, specifically 27 of $Diabetes\,Type\,1$ and 51 of $Diabetes\,Type\,2$; (iv) a total of 111 female (F) patients died by diabetes, specifically 43 of $Diabetes\,Type\,1$ and 68 of $Diabetes\,Type\,2$.

Similarly, consider the schema of the second co-occurrence dataset, defined as follows: $\{Country, \langle Gender, COUNT(Skin\,Cancer), COUNT(Lung\,Cancer), COUNT(Diabetes\,Type\,1), COUNT(Diabetes\,Type\,2)\rangle\}$. A possible instance is the following one: $\{France, \{\langle M\text{-}Cancer, 28, 61\rangle, \langle F\text{-}Cancer, 35, 74\rangle, \langle M\text{-}Diabetes, 30, 63\rangle, \langle F\text{-}Diabetes, 43, 68\rangle\}$, which the event that, in $France$, with *no* reference to the time, (i) a total of 89 male (M) patients died by cancer, specifically 28 of $Skin\,Cancer$ and 61 of $Lung\,Cancer$; (ii) a total of 109 female (F) patients died by cancer, specifically 35 of $Skin\,Cancer$ and 74 of $Lung\,Cancer$; (iii) a total of 93 male (M) patients died by diabetes, specifically 30 of $Diabetes\,Type\,1$ and 63 of $Diabetes\,Type\,2$; (iv) a total of 111 female (F) patients died by diabetes, specifically 43 of $Diabetes\,Type\,1$ and 68 of $Diabetes\,Type\,2$.

From the examples above, it should be explicitly noted that, in our co-occurrence dataset, we group-by the aggregate values of the target attributes by means of the values of the functional attributes (e.g., $F\text{-}Cancer$: aggregate values of $COUNT(Skin\,Cancer)$ and $COUNT(Lung\,Cancer)$ are grouped-by the gender of the patient F). This is due to the fundamental definition of co-occurrence analysis.

6.1.3 Multidimensional Aggregation

In the *Drill*-CODA multidimensional aggregation step, ad-hoc OLAP data cubes are built from the input co-occurrence datasets computed at the previous step (the co-occurrence analysis step). Given the input co-occurrence datasets $\mathcal{CO}_{\mathcal{S},CO} = \{C_{\mathcal{S},CO,0}, C_{\mathcal{S},CO,1}, \ldots, C_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}\}$, we compute $|\mathcal{A}_{\mathcal{S},CO}| - 1$ multidimensional OLAP data cubes as belonging to the set $\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO}) = \{DC_{\mathcal{S},CO,0}, DC_{\mathcal{S},CO,1}, \ldots, DC_{\mathcal{S},CO,|\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})|-1}\}$, where $|\mathcal{A}_{\mathcal{S},CO}| - 1 = |\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})| - 1$, such that each data cube $DC_{\mathcal{S},CO,k} \in \mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})$ is defined as follows:

$$\begin{aligned} DC_{\mathcal{S},CO,k} = \langle &\{A_{\mathcal{S},CO,0}, A_{\mathcal{S},CO,1}, \ldots, A_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}\}, \\ &\{AO_0(T_{S_j,0}), AO_1(T_{S_j,1}), \ldots, AO_{|\mathcal{T}_{S_j}|-1}(T_{S_j,|\mathcal{T}_{S_j}|-1})\}\rangle, \\ &\forall k \in \{0, 1, \ldots, |\mathcal{A}_{\mathcal{S},CO}| - 1\} \end{aligned} \tag{6.2}$$

such that: (i) $A_{\mathcal{S},CO,k}$, where $k \in \{0, 1, \ldots, |\mathcal{A}_{\mathcal{S},CO}| - 1\}$ denotes a dimension (which corresponds to a co-occurrence attribute – see Section 6.1.2); (ii) AO_z,

where $z \in \{0, 1, \ldots, |AO|-1\}$, denotes an aggregate operator selected from the set AO (see Section 6.1.1); (*iii*) T_{S_k}, where $k \in \{0, 1, \ldots, |\mathcal{T}_{S_j}| - 1\}$, denotes a target attribute of interest for the analysis (see Section 6.1.1). It should be noted, here, that: (*i*) each OLAP data cube $DC_{\mathcal{S},CO,k} \in \mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})$ is, formally, a *multiple-measure data cube*; (*ii*) the number of measures, which corresponds to the number of attributes of interest for the analysis, is the *same* for each OLAP data cube $DC_{\mathcal{S},CO,k} \in \mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})$.

To give an example, consider a simple two-dimensional model. Here, let $\langle \{Year, Gender\text{-}Disease\}, \{COUNT(\{Skin\,Cancer, Lung\,Cancer\}), COUNT(\{Diabetes\,Type\,1, Diabetes\,Type\,2\})\}\rangle$ be the schema of the first (two-dimensional) OLAP data cube. A possible data cube cell instance is the following one: $\langle 2020, M\text{-}Cancer\rangle = \langle 32, 69\rangle$, which models the event that, during 2020, with *no* reference to the location, a total number of 32 male (M) patient died by $Skin\,Cancer$ and a total number of 69 male (M) patient died by $Lung\,Cancer$.

Similarly, let $\langle \{Country, Gender\text{-}Disease\}, \{COUNT(\{Skin\,Cancer, Lung\,Cancer\}), COUNT(\{Diabetes\,Type\,1, Diabetes\,Type\,2\})\}\rangle$ be the schema of the second (two-dimensional) OLAP data cube. A possible data cube cell instance is the following one: $\langle Italy, F\text{-}Diabetes\rangle = \langle 31, 55\rangle$, which models the event that, in $Italy$, with *no* reference to the time, a total number of 31 female (F) patient died by $Diabetes\,Type\,1$ and a total number of 55 female (F) patient died by $Diabetes\,Type\,2$.

6.1.4 Drill-Across Querying

In the *Drill*-CODA drill-across querying step, given the collection of OLAP data cubes $\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO}) = \{DC_{\mathcal{S},CO,0}, DC_{\mathcal{S},CO,1}, \ldots, DC_{\mathcal{S},CO,|\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})|-1}\}$, computed at the previous step (the multidimensional aggregation step), we generate, for each data cube $DC_{\mathcal{S},CO,k} \in \mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})$, a *full-dimensional drill-across query* $Q_{\mathcal{Q},CO,k}$, defined as follows:

$$\begin{gathered} Q_{\mathcal{S},CO,k} = \langle \{[A_{\mathcal{S},CO,0}[0] : A_{\mathcal{S},CO,0}[|A_{\mathcal{S},CO,0}| - 1]], \\ [A_{\mathcal{S},CO,1}[0] : A_{\mathcal{S},CO,1}[|A_{\mathcal{S},CO,1}| - 1]] \ldots, \\ [A_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}[0] : A_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}[|A_{\mathcal{S},CO,|\mathcal{A}_{\mathcal{S},CO}|-1}| - 1]]\}, \\ AO_k(T_{S_j,k})\rangle, \forall k \in \{0, 1, \ldots, |\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})| - 1\} \end{gathered} \tag{6.3}$$

such that: (*i*) $A_{\mathcal{S},CO,k}$, where $k \in \{0, 1, \ldots, |\mathcal{A}_{\mathcal{S},CO}| - 1\}$ denotes a dimension of $DC_{\mathcal{S},CO,k}$ (which corresponds to a co-occurrence attribute – see Section 6.1.2); (*ii*) $A_{\mathcal{S},CO,k}[0]$ denotes the *first* dimensional member in $A_{\mathcal{S},CO,k}$; (*iii*) $A_{\mathcal{S},CO,k}[|A_{\mathcal{S},CO,k}| - 1]$ denotes the *last* dimensional member in $A_{\mathcal{S},CO,k}$; (*iv*) AO_z, where $z \in \{0, 1, \ldots, |AO| - 1\}$, denotes an aggregate operator selected from the set AO (see Section 6.1.1); (*v*) T_{S_k}, where $k \in \{0, 1, \ldots, |\mathcal{T}_{S_j}| - 1\}$, denotes a target attribute of interest for the analysis (see Section 6.1.1). It should be noted that the full-dimensional drill-across query $Q_{\mathcal{S},CO,k}$ spans *all* the dimensions of $DC_{\mathcal{S},CO,k}$ along *all* their dimensional domains.

By iterating the described procedure for each data cube $DC_{\mathcal{S},CO,k} \in \mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})$, we obtain the so-called *full-dimensional drill-across query set* $\mathcal{Q}_{\mathcal{CO}}(\mathcal{S}) = \{Q_{\mathcal{Q},CO,0}, Q_{\mathcal{Q},CO,1}, \ldots, Q_{\mathcal{Q},CO,|\mathcal{Q}_{\mathcal{CO}}(\mathcal{S})|-1}\}$. After that, each drill-across query $Q_{\mathcal{Q},CO,k} \in \mathcal{Q}_{\mathcal{CO}}(\mathcal{S})$ is executed against *all* the collection of OLAP data cubes $\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO}) = \{DC_{\mathcal{S},CO,0}, DC_{\mathcal{S},CO,1}, \ldots, DC_{\mathcal{S},CO,|\mathcal{DC}(\mathcal{CO}_{\mathcal{S},CO})|-1}\}$, thus finally originating the full-dimensional correlation set $\mathcal{D}_{\mathcal{CO}}(\mathcal{S})$. From Section 6.1.1, remind that $\mathcal{D}_{\mathcal{CO}}(\mathcal{S})$ stores collections of correlated aggregates.

To give an example, consider a simple two-dimensional model. Let

$$\begin{gathered}\langle\{Year, Gender\text{-}Disease\},\\ \{COUNT(\{Skin\,Cancer, Lung\,Cancer\}),\\ COUNT(\{\ Diabetes\,Type\,1, Diabetes\,Type\,2\})\}\rangle\end{gathered} \tag{6.4}$$

be the schema of the first (two-dimensional) OLAP data cube, and

$$\begin{gathered}\langle\{Country, Gender\text{-}Disease\},\\ \{COUNT(\ \{Skin\,Cancer, Lung\,Cancer\}),\\ COUNT(\{Diabetes\,Type\,1, Diabetes\ \ Type\,2\})\}\rangle\end{gathered} \tag{6.5}$$

be the schema of the second (two-dimensional) OLAP data cube, respectively. Let $\langle\{[2020 : 2023], [M\text{-}Cancer : F\text{-}Diabetes]\}, SUM\rangle$ be the input drill-across query against the two data cubes. The answer to the query is $\langle 358, 734\rangle$. The latter models the event that, from 2020 to 2023, a total number of 358 patients, with *no* reference to their sex, died by *Cancer* (including both *Skin Cancer* and *Lung Cancer*), and a total number of 734 patients, with *no* reference to their sex, died by *Diabetes* (including both *Diabetes Type* 1 and *Diabetes Type* 2).

6.2 *Drill*-CODA Cloud-Based Reference Architecture

In this Section, we introduce the Cloud-based reference architecture for the proposed *Drill*-CODA framework. We start by elucidating the underlying motivation for a real-world case study of our technique, by highlighting how *Drill*-CODA can be successfully used in the context of big data analytics platforms.

Modern big data analytics applications usually run on top of massive, large-scale big data repositories. As a consequence, there is a need for accessing, processing and analyzing such repositories via both well-consolidated big data management and analytics techniques and well-established Cloud-based big data processing platforms, such as *Hadoop*, *Spark* and *Kylin*.

In reply to these clear requirements, *Drill*-CODA must be deployed in a naive big data environment, as to take advantages from high-computation capabilities, scalability, virtualization, parallel/distributed executions, in-memory partial computations, and so forth. This evidence is stirred-up by the fact

Drill-CODA mostly processes multidimensional big data, hence it can easily incur in the so-called *curse of dimensionality* problem, meaning that performance of algorithms over multidimensional data decreases when the number of dimensions of input datasets increases. As a consequence, our study explores the anatomy and the functionalities of the big-data-aware *Drill*-CODA deployment.

In reply to these important requirements, Figure 6.2 should the Cloud-based *Drill*-CODA reference architecture.

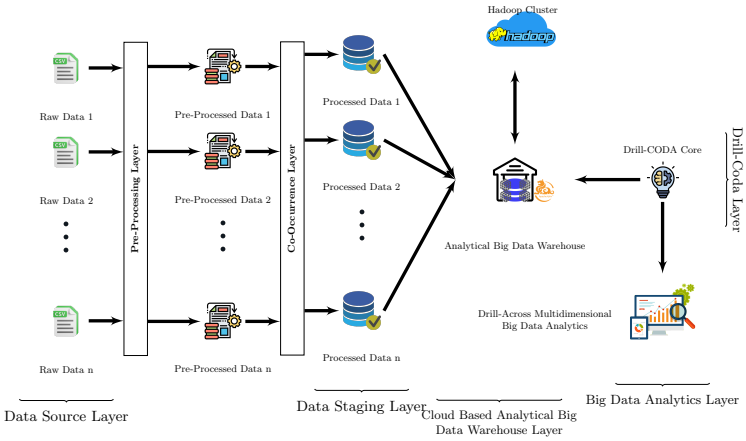


Fig. 6.2: The Cloud-Based *Drill*-CODA Reference Architecture

As shown in Figure 6.2, the Cloud-based *Drill*-CODA reference architecture includes the following layers:

1. ***Data Source Layer.*** In this layer, the original data sources of our Cloud-based *Drill*-CODA framework are fed as input to our enabling tool. Data, as collected from their sources (web, repositories, and so forth) are used as main entry for our data flow. Depending on their format and structure, that should be "unified" for subsequent processed, we apply cleansing and formatting transformations on them before considering them ready for the next data staging phase.
2. ***Pre-Processing Layer.*** Here, normalized data sources are pre-processed according to the *Drill*-CODA paradigm (see Section 6.1). This calls for a pre-processing step to cleanse and reformat data columns when needed, and above all, the crafting of data for the respective co-occurrence attributes, so that a valid drill-down operation could later be applied on the OLAP cubes to analyze. Also, aggregation along hierarchies is performed.
3. ***Co-Occurrence Layer.*** Here, the *Co-Occurrence Layer* supports our co-occurrence analysis (see Section 6.1). Our main goal through this phase is

to ensure that co-occurrence attributes are present and allow the creation of a consequent hierarchy later-on for our multidimensional analysis. The co-occurrence aggregate data are provided as final output.

4. ***Data Staging Layer.*** In this layer, we materialize the co-occurrence data in suitable data structures, on top of which multidimensional analysis is later performed. This step is required to prepare the data for querying in highly-multidimensional fashion and make the data (type and format essentially) suitable for deployment onto the *data warehouse solutions*.
5. ***Cloud-Based Analytical Big Data Warehouse Layer.*** In this layer, thanks to the *Kylin* OLAP framework, and to its interoperability with *Hadoop*, multidimensional data are aggregated on top of staging co-occurrence data in a *MapReduce* fashion.
6. ***Drill-CODA Layer.*** In the *Drill-CODA Layer*, the core components of *Drill*-CODA run in order to derive drill-across multidimensional big data analytics over big co-occurrence aggregate hierarchical data, according to the main guidelines proposed by our research (see Section 6.1).
7. ***Big Data Analytics Layer.*** Here, the final desiderata big data analytics applies, in order to provide useful and actionable knowledge from large-scale big data repositories, mostly by focusing the attention on the full-dimensional correlation pattern discovery

6.3 Experimental Evaluation and Analysis

In order to assess our big data analytics framework, we introduced the following metrics, for two different classes of experiments. In the first class, we analyzed COUNT aggregations over the aggregate multidimensional co-occurrence data. In the second class, we performed correlation analysis on the aggregate multidimensional co-occurrence data. In particular, for the latter analysis, the goal was to analyze the correlation of these type of data. Specifically, we chose to exploit two well-known correlation indexes, namely the Pearson correlation index [175] and the Spearman correlation index [176]. In fact, those indexes are the state-of-the-art references for correlation analysis. While Spearman focuses on the ranges of the data distribution, so that it perfectly captures the scope of aggregation analysis, Pearson focuses on the effective values (i.e., the magnitudes) of data, still capturing other properties of aggregation analysis. Indeed, in aggregations, we are usually interested both into the magnitude values and the ranges on which those aggregations are computed. We use in Drill-CODA because of in Drill-CODA we specifically deal with data aggregations. More into details, for the correlation analysis we built the correlation heatmap that reports the degree of correlation for pairs of target attributes. Three experiments, using six different real-life datasets were implemented.

6.3.1 Experiment 1: Substance Use and Narcan Administration

. In the first experiment, we focused on the substance use in Winnipeg, MB, Canada, which we correlate with a counter-effect substance known as Narcan. *Datasets* Mainly, we used the following real-life datasets:

- **Substance Use**: This dataset describes events involving Alcohol, Cocaine, Crystal Meth, Marijuana or Opioids use as reported by the *Winnipeg Fire Paramedic Service* (WFPS) paramedic crew [129].
- **Narcan Administration**: This dataset stores incidents involving Narcan (Naloxone) administration to a patient once or more [128].

Co-Occurence plots Plots of Time and Location co-occurrence for different events depending on the context of the dataset were derived from the raw input, below we display the finding for all 3 experiences that were generated using *Python/Matplotlib* library. First, let us notice that co-occurence data is plotted in an anonymized manner, since only the *Year* or *Region* attribute numbers are depicted (those attributes being the higher level of, respectively, the time and the location hierarchies).

As shown in 6.3, the first experiment on time co-occurence, shows a substantial increase of substance uses across time until the year 2021, in 2022 a low substance use numbers were registered, for 2023 though, yet incomplete data explains the low counts. In addition to the mentioned trend, a simultaneous steady increase in both Narcan and other substances is noticeable especially from year 2016 (surge year) to 2022 (decline year). Whereas, based on locations 6.4, the substance use is quasi-constant, meaning that all locations had, roughly, used the same quantities of substances.

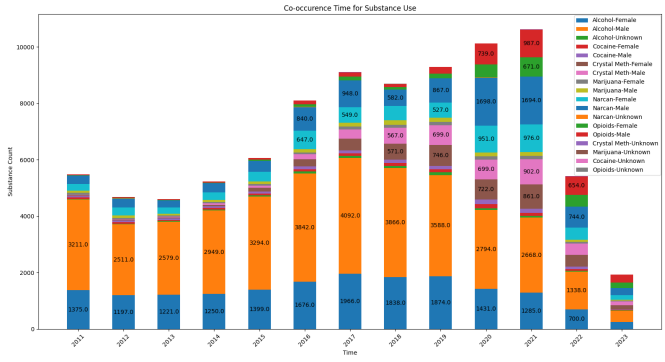


Fig. 6.3: Experiment 1 Time Co-Occurent Stacked Bars Plot

Correlation plots

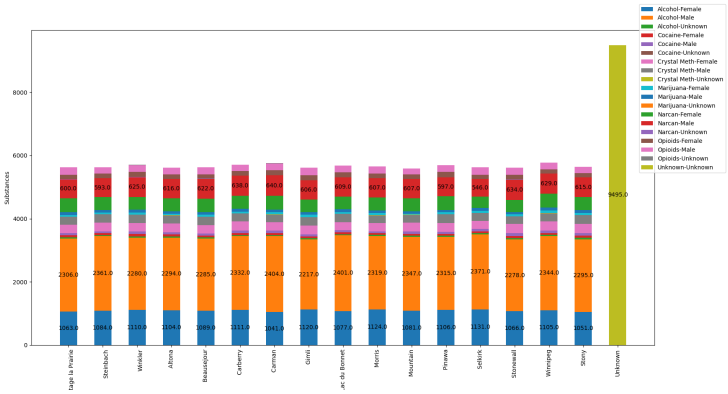


Fig. 6.4: Experiment 1 Location Co-Occurent Stacked Bars Plot

Figure 6.5 and Figure 6.6 show the full-dimensional Pearson correlation heatmap and the full-dimensional Spearman correlation heatmap for the first experiment, respectively.

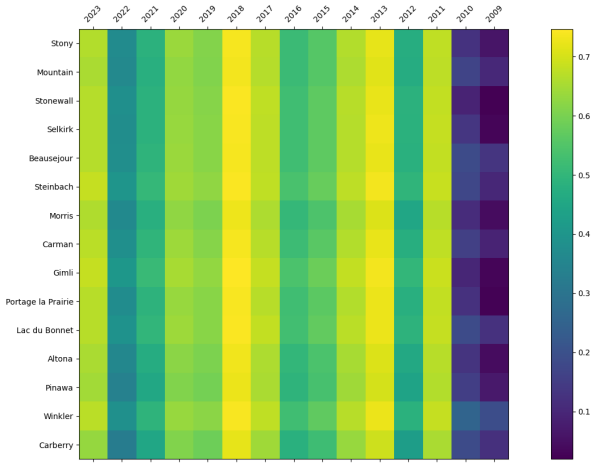


Fig. 6.5: Experiment 1 Full-Dimensional Pearson Correlation Heatmap

As mentioned, after analyzing the cubes and extracting relevant information, we proceed with the drill-across 'step' in order to eventually derive correlations of pearsons and spearman. For experiment 1, both pearson and

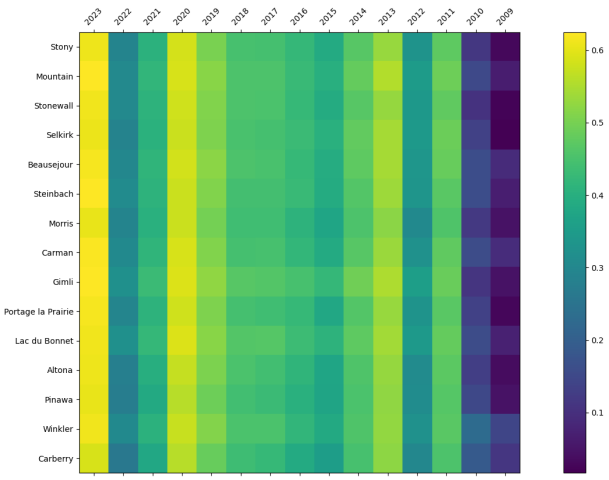


Fig. 6.6: Experiment 1 Full-Dimensional Spearman Correlation Heatmap

spearman heatmaps show a pronounced correlation between the two considered data cubes for years 2013, 2018 and 2023. (yellow colored lines), while the weakest correlation were spotted for years 2009 and 2010 (indigo colored lines). It is also worth noticing, that the correlation is heavily influenced by the years than by the locations.

6.3.2 Experiment 2: Diabetes and Cancer Deaths

In the second experiment, we considered diabetes statistics as its relation with the cancer deaths. *Datasets* Here, we used the following real-life datasets.

- **Diabetes**: The dataset includes age-standardized estimates of the prevalence of diabetes and its associated risk factors. Data are gathered from 200 countries, while the range of time is from 1980 to 2014 [21].
- **Cancer Deaths**: This dataset stores cancer deaths worldwide over the last 29 years (1990-2019) [104].

Co-Occurence plots

This expriment showed higher number of deaths for both diabetes and cancer in Asia & Pacific region as well as in Europe as depicted in Figures 6.7 and 6.8. The African continent had the lowest numbers of deaths by the two considered illnesses. When it comes to the time co-occurence, experiment 2 shows a lightly fluctuanting numbers while diabetes deaths among both genders is much more important than liver, kidney or tracheal, bronchus and lung cancer deaths.

Correlation plots

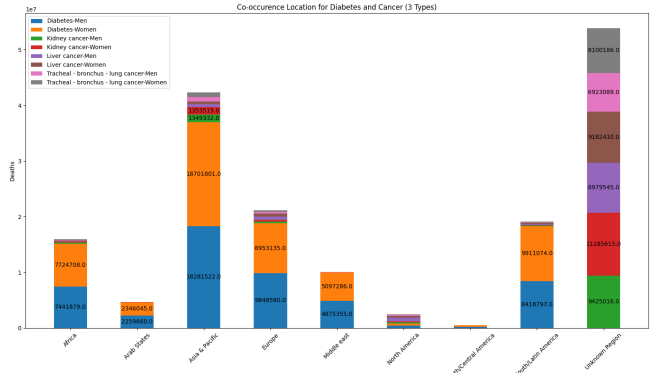


Fig. 6.7: Experiment 2 Location Co-Occurent Stacked Bars Plot

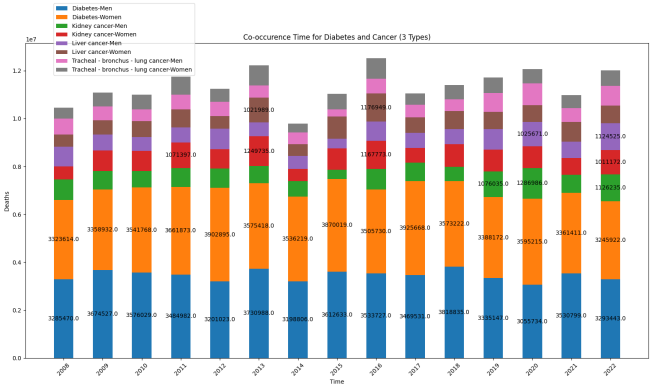


Fig. 6.8: Experiment 2 Time Co-Occurent Stacked Bars Plot

For this experiment, we observe from the obtained results, that no particular patterns could be devised for the Pearson heatmap but a low correlation for Arab States as well as for North America, based on low values for years 2011, 2012 and 2020. Also. a relatively high correlation is shown by the case for the South/Central America region. On the other hand, for the Spearman heatmap, an overall lower correlation values are shown for the 2008-2013 yearly range. Associated Figures are shown in 6.10 and 6.9.

6.3.3 Experiment 3: Cancer Incidence and Mental Disorders

In the third experiment, we focused on cancer incidence as related to mental disorders. *Datasets* Here, we used the following real-life datasets:

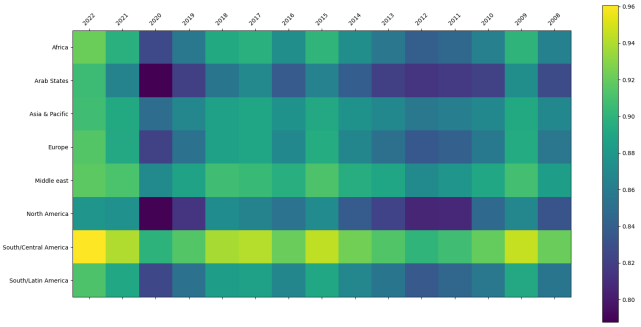


Fig. 6.9: Experiment 2 Full-Dimensional Pearson Correlation Heatmap

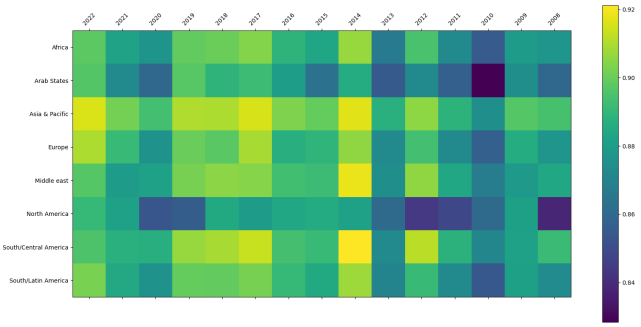


Fig. 6.10: Experiment 2 Full-Dimensional Spearman Correlation Heatmap

- **Cancer Incidence (CI5Plus)**: The *CI5Plus* database contains updated annual incidence rates for 124 selected populations from 108 cancer registries published in *CI5Plus*, for the longest period available (up to 2012), for all cancers and 28 major types [201].
- **Mental Disorders**: This dataset contains informative data from Countries across the globe about the prevalence of mental health disorders, including schizophrenia, bipolar disorder, eating disorders, anxiety disorders, drug use disorders, depression and alcohol use disorders [76].

Co-Occurence Plots In this experiment three (Figures 6.11 and 6.12), higher number of cancer and mental disorders were still registered in Asia & pacific and Europe regions, while the African continent had low numbers of incidence related to the considered health diseases. Particularly for the time

co-occurence, a spike for cancer incidence is noticeable starting from year 1998 while mental disorders counting was highly fluctuating for both men and women.

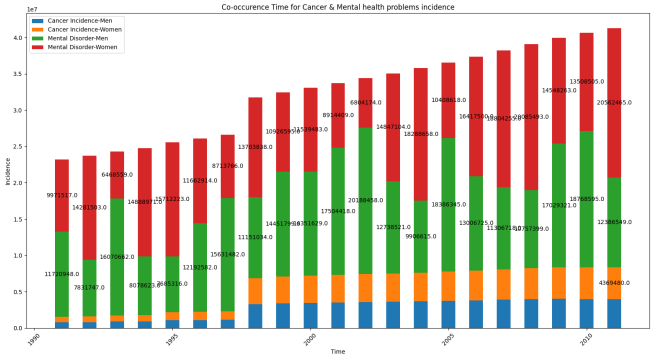


Fig. 6.11: Experiment 3 Time Co-Occurent Stacked Bars Plot

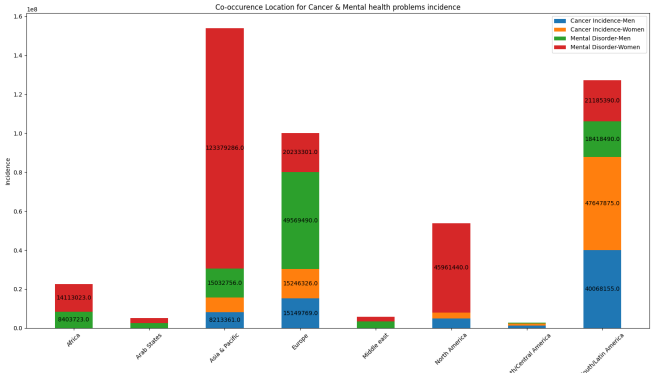


Fig. 6.12: Experiment 3 Location Co-Occurent Stacked Bars Plot

Correlation plots Here, both Pearson and Spearman heatmaps demonstrate low correlation indexes for both South/Central and Latin America. In addition to this, by deeply investigating the Spearman heatmap, a pattern is shown for the years from 1990 to 1997 for all regions except South/Central and Latin America. Figures are shown in 6.13 and 6.14.

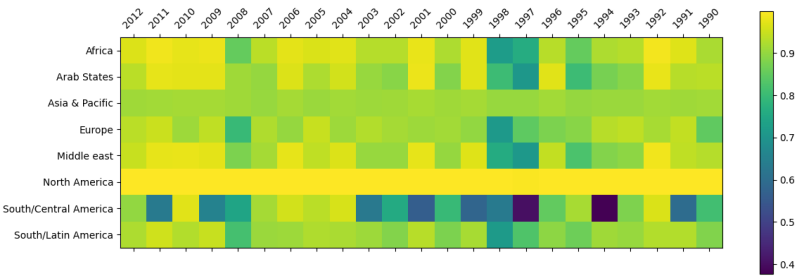


Fig. 6.13: Experiment 3 Full-Dimensional Pearson Correlation Heatmap

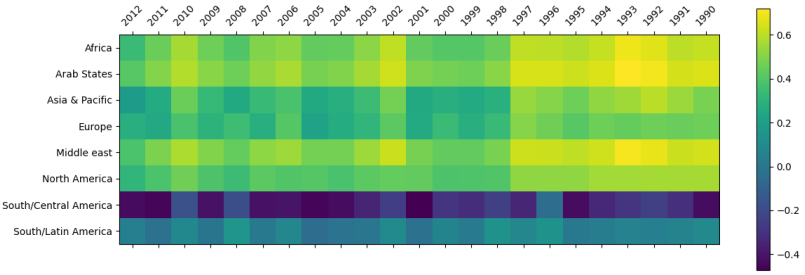


Fig. 6.14: Experiment 3 Full-Dimensional Spearman Correlation Heatmap

6.4 Summary

We focused on the co-occurrence events data to privately derive statistical outputs such as correlation from diverse sources of OLAP cubes. The goal being: given several OLAP cubes as input presenting events-based data, and defining attributes of location and time, to privately derive analytics through using a parallel processing methodology of the involved cubes. We tackled correlation based statistical analysis of such cubes. Such analytics are thus derived from a multitude of OLAP cubes to enable the co-occurrence aspect of the analysis. A cloud based architecture was also presented to cover the scalability issues that might stem from executing such technique over potentially large cubes.

7

Conclusions

Privacy preserving data analytics topic encompasses the privacy techniques that serve the analytical process especially when the data are sensitive ergo needs to be protected. Proposals in this topic have seen a boom in number over the last decade due to a myriad of reasons mainly related to the complexity of processing data of high amount that are diverse and generated at high paces. In fact, it has been established that there's a trade-off between data utility and data privacy which is in most cases unbalanced towards data privacy as data standards and regulations dictate. This limits the potential of data analytics in providing a comprehensive analysis over the target data. It is explained mainly by the many ways and approaches data privacy methods and techniques tackle data to achieve the expected privacy results to obey to data sharing constraints.

In this thesis, we have presented privacy-preserving data analytical solutions that attempt to solve the previous current stated limitations. A proper assessment and evaluation of our frameworks, models and techniques has also been described. Basically, we aimed at providing scalable, effective and efficient solutions to tackle state-of-the-art challenges stemming mainly from the massive growth of generated data each day while maximizing the benefits of data analytics. We have argued how our solutions make good fit for current state of the art literature proposals and solutions and how our defined models and frameworks could enhance the outcome of the data analytical process while complying to specific data safety constraints. Finally, the contribution made in each chapter are the following:

- A framework called AB-DOM that enables data privacy via innovative data model that accounts for data diversity all in data lake contexts.
- A system called QFLS that enables, in distributed environments, privacy preserving data analytics through data locality.
- A technique named Drill-CODA that, in the data warehouse context, that enables distributed drill-across querying in privacy preserving manner as to yield data analytics, namely, correlation-based.

References

[1] https://www.gartner.com/.

[2] URL: https://hadoop.apache.org/.

[3] URL: https://hadoop.apache.org/.

[4] URL: https://spark.apache.org/.

[5] URL: https://archive.ics.uci.edu/ml/datasets/Immunotherapy+Dataset.

[6] Afsoon Abbasi and Behnaz Mohammadi. "A clustering-based anonymization approach for privacy-preserving in the healthcare cloud". In: *Concurrency and Computation: Practice and Experience* 34.1 (2022), e6487. DOI: https://doi.org/10.1002/cpe.6487. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6487. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6487.

[7] Umar Abdulkadir et al. "Ring Learning With Error-Based Encryption Scheme for the Privacy of Electronic Health Records Management". In: *2022 5th Information Technology for Education and Development (ITED)*. 2022, pp. 1–5. DOI: 10.1109/ITED56637.2022.10051260.

[8] A. A. Abdullah, M. M. Hassan, and Y. T. Mustafa. "A Review on Bayesian Deep Learning in Healthcare: Applications and Challenges". In: *IEEE Access 10* (2022), pp. 36538–36562.

[9] Charu Aggarwal. "On Randomization, Public Information and the Curse of Dimensionality". In: May 2007, pp. 136–145. ISBN: 1-4244-0803-2. DOI: 10.1109/ICDE.2007.367859.

[10] Charu C. Aggarwal. "On K-Anonymity and the Curse of Dimensionality". In: *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB '05. Trondheim, Norway: VLDB Endowment, 2005, pp. 901–909. ISBN: 1595931546.

[11] Gagan Aggarwal et al. "Achieving anonymity via clustering". In: *ACM Transactions on Algorithms (TALG)* 6.3 (2010), pp. 1–19.

[12] Rakesh Agrawal, Ramakrishnan Srikant, and Dilys Thomas. "Privacy preserving OLAP". In: June 2005, pp. 251–262. DOI: 10.1145/1066157.1066187.

[13] Riaz Ahmed, Sumayya Shaheen, and Simon P. Philbin. "The role of big data analytics and decision-making in achieving project success". In: *Journal of Engineering and Technology Management* 65 (2022), p. 101697. ISSN: 0923-4748. DOI: https://doi.org/10.1016/j.jengtecman.2022.101697. URL: https://www.sciencedirect.com/science/article/pii/S0923474822000273.

[14] Jayakrishnan Ajayakumar and Kambiz Ghazinour. "I am at home: Spatial Privacy Concerns with Social Media Check-ins". In: *Procedia Computer Science* 113 (2017). The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017)

/ Affiliated Workshops, pp. 551–558. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.08.278. URL: https://www.sciencedirect.com/science/article/pii/S1877050917316873.

[15] Giebler et al. " The Data Lake Architecture Framework: A Foundation for Building a Comprehensive Data Lake Architecture". In: 2021.

[16] Nambiar et al. "An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management". In: vol. 6. Nov. 2022, p. 132. DOI: 10.3390/bdcc6040132.

[17] Suriarachchi et al. "Provenance as Essential Infrastructure for Data Lakes". In: June 2016, pp. 178–182. ISBN: 978-3-319-40592-6. DOI: 10.1007/978-3-319-40593-3_16.

[18] Yang et al. "Analysis of Data Warehouse Architectures: Modeling and Classification". In: Jan. 2019, pp. 604–611. DOI: 10.5220/0007728006040611.

[19] Gaudenz Alder. *mxGraph*. https://jgraph.github.io/mxgraph/. 2005.

[20] Krall Alexander, Finke Daniel, and Yang Hui. "Mosaic Privacy-Preserving Mechanisms for Healthcare Analytics". In: *IEEE Journal of Biomedical and Health Informatics* 25.6 (2021), pp. 2184–2192. DOI: 10.1109/JBHI.2020.3036422.

[21] Wajahat Ali. *Diabetes*. 2023. URL: https://www.kaggle.com/code/wawajahat/diabetes-dataset-by-age-standardized.

[22] Zenab Amin et al. "Preserving Privacy of High-Dimensional Data by l-Diverse Constrained Slicing". In: *Electronics* 11.8 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11081257. URL: https://www.mdpi.com/2079-9292/11/8/1257.

[23] J. Archenaa and Mary Anita. "A Survey of Big Data Analytics in Healthcare and Government". In: *Procedia Computer Science* 50 (Dec. 2015), pp. 408–413. DOI: 10.1016/j.procs.2015.04.021.

[24] Claudio Ardagna et al. "Location Privacy Protection Through Obfuscation-Based Techniques". In: vol. 4602. July 2007, pp. 47–60. ISBN: 978-3-540-73533-5. DOI: 10.1007/978-3-540-73538-0_4.

[25] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: vol. 8. Jan. 2007, pp. 1027–1035. DOI: 10.1145/1283383.1283494.

[26] Farzindar Atefeh and Wael Khreich. "A Survey of Techniques for Event Detection in Twitter". In: *Comput. Intell.* 31.1 (Feb. 2015), pp. 132–164. ISSN: 0824-7935. DOI: 10.1111/coin.12017. URL: https://doi.org/10.1111/coin.12017.

[27] Piero Baraldi et al. "Reconstruction of missing data in multidimensional time series by fuzzy similarity". In: *Applied Soft Computing* 26 (Jan. 2015). DOI: 10.1016/j.asoc.2014.09.038.

[28] R. J. Bayardo and R. Agrawal. "Data Privacy through Optimal K-Anonymization". In: 21st IEEE International Conference on Data Engineering. 2005, pp. 217–228.

[29] S. H. Begum and F. Nausheen. "A Comparative Analysis of Differential Privacy vs Other Privacy Mechanisms for Big Data". In: 2nd International Conference on Inventive Systems and Control. 2018, pp. 512–516.

[30] Y.a Benkaouz, M.a Erradi, and B.b Freisleben. "Distributed privacy-preserving data aggregation via anonymization". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9466 (2015). cited By 0, pp. 94–108. DOI: 10.1007/978-3-319-26850-7_7. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84961141090%5C&doi=10.1007%5C%2f978-3-319-26850-7_7%5C&partnerID=40%5C&md5=518bf129894b27ee29b635e53f9baf0c.

[31] Elisa Bertino, Igor Nai Fovino, and Loredana Parasiliti Provenza. "A Framework for Evaluating Privacy Preserving Data Mining Algorithms". In: *Data Mining and Knowledge Discovery* 11.2 (2005), pp. 121–154.

[32] Mario Bochicchio, Alfredo Cuzzocrea, and Lucia Vaira. "A Big Data Analytics Framework for Supporting Multidimensional Mining over Big Healthcare Data". In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2016, pp. 508–513. DOI: 10.1109/ICMLA.2016.0090.

[33] Alexander Bogdanov et al. "Big Data Virtualization: Why and How". In: *CEUR Workshop Proceedings (2679)*. 2020, pp. 11–21.

[34] Ji-Won Byun et al. "Efficient k-Anonymization Using Clustering Techniques". In: *Advances in Databases: Concepts, Systems and Applications, 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007, Proceedings*. Vol. 4443. Lecture Notes in Computer Science. Springer, 2007, pp. 188–200.

[35] Cristian Cadar et al. "Data Randomization". In: (Jan. 2008).

[36] Jesus Camacho-Rodriguez et al. "Apache Hive: From MapReduce to Enterprise-grade Big Data Warehousing". In: *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*. ACM, 2019, pp. 1773–1786.

[37] Jianneng Cao et al. "SABRE: a Sensitive Attribute Bucketization and REdistribution framework for". In: *VLDB J.* 20 (Feb. 2011), pp. 59–81. DOI: 10.1007/s00778-010-0191-9.

[38] Centers for Medicare & Medicaid Services. "The Health Insurance Portability and Accountability Act of 1996 (HIPAA)". In: (1996).

[39] Weng Chao et al. "Benefits and Challenges of Electronic Health Record System on Stakeholders: A Qualitative Study of Outpatient Physicians". In: *Journal of medical systems* 37 (Aug. 2013), p. 9960. DOI: 10.1007/s10916-013-9960-5.

[40] Ritu Chauhan, Harleen Kaur, and Victor Chang. "An Optimized Integrated Framework of Big Data Analytics Managing Security and Privacy in Healthcare Data". In: *Wireless Personal Communications* 117 (Mar. 2021), pp. 1–22. DOI: 10.1007/s11277-020-07040-8.

[41] Jing Chen. "Multidimensional analysis model of agricultural product supply chain competition based on mean fuzzy". In: *J. Intell. Fuzzy Syst.* 41.2 (2021), pp. 3591–3602. DOI: 10.3233/JIFS-210962. URL: https://doi.org/10.3233/JIFS-210962.

[42] Mandy Chessell. *Governing and Managing Big Data for Analytics and Decision Makers*. RedBooks, 2014.

[43] Lin Chi et al. "Differential Privacy Preserving in Big Data Analytics for Connected Health". In: *Journal of Medical Systems* 40 (Feb. 2016). DOI: 10.1007/s10916-016-0446-0.

[44] Shirley Coleman et al. "How Can SMEs Benefit from Big Data? Challenges and a Path Forward". In: *Quality and Reliability Engineering International* 32.6 (2016), pp. 2151–2164. DOI: https://doi.org/10.1002/qre.2008. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre.2008. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.2008.

[45] Diego Collarana, Christoph Lange, and S. Auer. "FuhSen: A Platform for Federated, RDF-based Hybrid Search". In: *Proceedings of the 25th International Conference Companion on World Wide Web* (2016).

[46] Gregory W. Corder and Dale I. Foreman. *Nonparametric Statistics: A Step-by-Step Approach*. Wiley, 2014.

[47] H. Cramér. *"Random Variables and Probability Distributions"*. Cambridge University Press, 2004.

[48] Claudivan Cruz Lopes et al. "Processing OLAP Queries over an Encrypted Data Warehouse Stored in the Cloud". In: Sept. 2014. ISBN: 978-3-319-10159-0. DOI: 10.1007/978-3-319-10160-6_18.

[49] Louis Cuny. *Jquery multi-select*. https://github.com/lou/multi-select/. 2011.

[50] Elizabeth H. Cuthill and John M. McKee. "Reducing the bandwidth of sparse symmetric matrices". In: *ACM '69*. 1969. URL: https://api.semanticscholar.org/CorpusID:18143635.

[51] A. Cuzzocrea. "Big Data Lakes: Models, Frameworks, and Techniques". In: IEEE International Conference on Big Data and Smart Computing. 2021, pp. 1–4.

[52] Alfredo Cuzzocrea and Elisa Bertino. "Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach". In: *Journal of Computer and System Sciences* 77.6 (2011), pp. 965–987. ISSN: 0022-0000. DOI: https://doi.org/10.1016/j.jcss.2011.02.004.

[53] Alfredo Cuzzocrea, Elisa Bertino, and Domenico Saccà. "Towards a theory for Privacy Preserving Distributed OLAP". In: *ACM Inter-*

national Conference Proceeding Series (Mar. 2012). DOI: 10.1145/2320765.2320826.

[54] Alfredo Cuzzocrea, Elisa Bertino, and Domenico Saccà. "Towards a theory for Privacy Preserving Distributed OLAP". In: *ACM International Conference Proceeding Series* (Mar. 2012). DOI: 10.1145/2320765.2320826.

[55] Alfredo Cuzzocrea, Filippo Furfaro, and Domenico Sacca. "Enabling OLAP in Mobile Environments via Intelligent Data Cube Compression Techniques". In: *Journal of Intelligent Information Systems* 33.2 (2009), pp. 95–143.

[56] Alfredo Cuzzocrea, Vincenzo Russo, and Domenico Saccà. "A Robust Sampling-Based Framework for Privacy Preserving OLAP". In: *Data Warehousing and Knowledge Discovery*. Ed. by Il-Yeol Song, Johann Eder, and Tho Manh Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 97–114. ISBN: 978-3-540-85836-2.

[57] Alfredo Cuzzocrea and Domenico Saccà. "A Theoretically-Sound Accuracy/Privacy-Constrained Framework for Computing Privacy Preserving Data Cubes in OLAP Environments". In: Sept. 2012, pp. 527–548. ISBN: 978-3-642-33614-0. DOI: 10.1007/978-3-642-33615-7_6.

[58] Alfredo Cuzzocrea and Paolo Serafino. "LCS-Hist: Taming Massive High-Dimensional Data Cube Compression". In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. EDBT '09. Saint Petersburg, Russia: Association for Computing Machinery, 2009, pp. 768–779. ISBN: 9781605584225. DOI: 10.1145/1516360.1516448.

[59] Alfredo Cuzzocrea and Selim Soufargi. "An Algorithmic Framework for Supporting Privacy-Preserving Big Data Publishing in Big Data Lakes". In: *IEEE Transactions on Big Data* (2023). Currently under revision.

[60] Alfredo Cuzzocrea and Selim Soufargi. "Drill-CODA: A Framework for Supporting Drill-Across Multidimensional Big Data Analytics over Big Co-Occurrence Aggregate Hierarchical Data". In: Accepted at DATA 2024.

[61] Alfredo Cuzzocrea and Selim Soufargi. "Privacy-Preserving Multidimensional Big Data Analytics Models, Methods and Techniques over Big Data: A Comprehensive Survey". In: *Expert Systems With Applications* (2023). Currently under revision.

[62] Alfredo Cuzzocrea and Selim Soufargi. "QFLS: A Cloud-Based Framework for Supporting Big Healthcare Data Management and Analytics from Big Data Lakes: Definitions, Requirements, Models and Techniques". In: *Proceedings of the 12th International Conference on Data Science, Technology and Applications, DATA 2023, Rome, Italy, July 11-13,2023*. SCITEPRESS, 2023, pp. 422–428.

[63] Alfredo Cuzzocrea and Selim Soufargi. "QFLS: A Complex Federated Big Data Analytics Learning System over Big Healthcare Data". In: *Big Data Research Journal* (2024). Currently under revision.

[64] Alfredo Cuzzocrea and Selim Soufargi. "Supporting Big Healthcare Data Management and Analytics: The Cloud-Based QFLS Framework". In: *Big Data Analytics and Knowledge Discovery - 25th International Conference,DaWaK 2023, Penang, Malaysia, August 28-30, 2023, Proceedings*. Vol. 14148. Lecture Notes in Computer Science. Springer, 2023, pp. 372–379.

[65] Alfredo Cuzzocrea et al. "F-TBDA: A Frequency-Based Temporal Big Data Analytics Technique for Mining and Analyzing Quality-Of-Life Indicators of Cancer Patients". In: *2023 IEEE International Conference on Big Data (BigData)*. 2023, pp. 5197–5205. DOI: 10.1109/BigData59044.2023.10386767.

[66] Alfredo Cuzzocrea et al. "OLAP over Big COVID-19 Data: A Real-Life Case Study". In: *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. 2022, pp. 1–6. DOI: 10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9927803.

[67] Alfredo Cuzzocrea et al. "Privacy-Preserving OLAP-Based Monitoring of Data Streams: The PP-OMDS Approach". In: *Proceedings of the 27th Italian Symposium on Advanced Database Systems, Castiglione della Pescaia (Grosseto), Italy, June 16-19, 2019*. Ed. by Massimo Mecella, Giuseppe Amato, and Claudio Gennaro. Vol. 2400. CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2400/paper-34.pdf.

[68] Alfredo Cuzzocrea et al. "Scaling Posterior Distributions over Differently-Curated Datasets: A Bayesian-Neural-Networks Methodology". In: *Foundations of Intelligent Systems*. Cham: Springer International Publishing, 2022, pp. 198–208. ISBN: 978-3-031-16564-1.

[69] Alfredo Cuzzocrea et al. "Supporting Privacy-Preserving Big Data Analytics on Temporal Open Big Data". In: *Procedia Computer Science* 198 (2022). 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare, pp. 112–121. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.12.217. URL: https://www.sciencedirect.com/science/article/pii/S187705092102456X.

[70] Alfredo Cuzzocrea et al. "The Emerging Challenges of Big Data Lakes, and a Real-Life Framework for Representing, Managing and Supporting Machine Learning on Big Arctic Data". In: *Advances in Intelligent Networking and Collaborative Systems*. Ed. by Leonard Barolli and Hi-

royoshi Miwa. Cham: Springer International Publishing, 2022, pp. 161–174. ISBN: 978-3-031-14627-5.

[71] Stefano Mazzocchi Daniel Ramage. *Federated Analytics: Collaborative Data Science without Data Collection*. https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html, 2020.

[72] A. Darrel et al. "The Benefits of Big Data Analytics in the Healthcare Sector: What Are They and Who Benefits?" In: *Big Data Analytics in Bioinformatics and Healthcare* (2015), pp. 406–439.

[73] Sabrina De Capitani di Vimercati et al. "Scalable Distributed Data Anonymization for Large Datasets". In: *IEEE Transactions on Big Data* (2022), pp. 1–14. DOI: 10.1109/TBDATA.2022.3207521.

[74] J. Dean and S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters". In: *Communications of the ACM 51(1)* (2008), pp. 107–113.

[75] Z. Dehghani and M. Fowler. *Data Mesh: Delivering Data-driven Value at Scale*. O'Reilly Media, 2022. ISBN: 9781492092391. URL: https://books.google.it/books?id=M5J5zgEACAAJ.

[76] The Devastator. *Mental Disorder*. 2023. URL: https://www.kaggle.com/datasets/thedevastator/uncover-global-trends-in-mental-health-disorder.

[77] A. Dhillon and A. Singh. "Machine Learning in Healthcare Data Analysis: A Survey". In: *Journal of Biology and Today's World 8(6)* (2019), pp. 1–10.

[78] J. Domingo-Ferrer, K. Muralidhar, and M. Bras-Amorós. "General Confidentiality and Utility Metrics for Privacy-Preserving Data Publishing Based on the Permutation Model". In: *IEEE Transactions on Dependable and Secure Computing 18(5)* (2020), pp. 2506–2517.

[79] Doreswamy and K. S. Harishkumar. "Multidimensional Data Model for Air Pollution Data Analysis". In: *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018, Bangalore, India, September 19-22, 2018*. IEEE, 2018, pp. 1684–1689. DOI: 10.1109/ICACCI.2018.8554621.

[80] Troy Bryan Downing. *Java RMI: Remote Method Invocation*. IDG Books Worldwide, Inc., 1998.

[81] Petros Drineas, Ravindran Kannan, and Michael Mahoney. "Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication". In: *SIAM J. Comput.* 36 (Jan. 2006), pp. 132–157. DOI: 10.1137/S0097539704442684.

[82] C. Dwork. "Differential Privacy". In: *In:* 33 (2006), pp. 1–12.

[83] H. Ebadi, T. Antignac, and D. Sands. "Sampling and Partitioning for Differential Privacy". In: *14th Annual Conference on Privacy, Security and Trust*. 2016, pp. 664–673.

[84] Ciceri Eleonora et al. "PAPAYA A platform for privacy preserving data analytics". In: *ERCIM News* 118 (2019).

[85] Nada Elgendy and Ahmed Elragal. "Big Data Analytics in Support of the Decision Making Process". In: *Procedia Computer Science* 100 (Jan. 2016), pp. 1071–1084. ISSN: 1877-0509. URL: https://www.sciencedirect.com/science/article/pii/S1877050916324206.

[86] Ehab ElSalamouny and Catuscia Palamidessi. *Reconstruction of the distribution of sensitive data under free-will privacy*. 2022. arXiv: 2208.11268 [cs.CR].

[87] Simone Fanelli et al. "Big data analysis for decision-making processes: challenges and opportunities for the management of health-care organizations". In: *Management Research Review* 46.3 (May 2022), pp. 369–389. DOI: 10.1108/MRR-09-2021-0648. URL: https://ideas.repec.org/a/eme/mrrpps/mrr-09-2021-0648.html.

[88] Huang Fang. "Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem". In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2015, pp. 820–824. DOI: 10.1109/CYBER.2015.7288049.

[89] Franziska Franke and Martin Hiebl. "Big data and decision quality: The role of management accountants' data analytics skills". In: *International Journal of Accounting and Information Management* 31 (Feb. 2023), pp. 93–127. DOI: 10.1108/IJAIM-12-2021-0246.

[90] Somchart Fugkeaw et al. "Developing Access Control Model of Web OLAP over Trusted and Collaborative Data Warehouses". In: Mar. 2010, pp. 393–413. ISBN: 978-1-84996-073-1. DOI: 10.1007/978-1-84996-074-8_15.

[91] B. C. Fung et al. "Privacy-Preserving Data Publishing: A Survey of Recent Developments". In: *ACM Computing Surveys 42(4)* (2010), pp. 1–53.

[92] Johannes Gehrke, Edward Lui, and Rafael Pass. "Towards Privacy for Social Networks: A Zero-Knowledge Based Definition of Privacy". In: Mar. 2011, pp. 432–449. ISBN: 978-3-642-19570-9. DOI: 10.1007/978-3-642-19571-6_26.

[93] Hemant Ghayvat et al. "CP-BDHCA: Blockchain-Based Confidentiality-Privacy Preserving Big Data Scheme for Healthcare Clouds and Applications". In: *IEEE Journal of Biomedical and Health Informatics* PP (July 2021), pp. 2168–2194. DOI: 10.1109/JBHI.2021.3097237.

[94] Gabriel Ghinita, Panos Kalnis, and Yufei Tao. "Anonymous Publication of Sensitive Transactional Data". In: *IEEE Transactions on Knowledge and Data Engineering* 23.2 (2011), pp. 161–174. DOI: 10.1109/TKDE.2010.101.

[95] Corinna Giebler et al. "The Data Lake Architecture Framework: a Foundation for Building a Comprehensive Data Lake Architecture". In: *Conference for Database Systems for Business, Technology and Web (BTW)*. 2021, pp. 351–370.

[96] Jim Gray et al. "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub Totals". In: *Data Min. Knowl. Discov.* 1.1 (1997), pp. 29–53.

[97] Jenniffer Sidney Guerrero-Prado, Wilfredo Alfonso-Morales, and Eduardo Caicedo Bravo. "A Data Analytics/Big Data Framework for Advanced Metering Infrastructure Data". In: *Sensors (Basel, Switzerland)* 21 (2021). URL: https://api.semanticscholar.org/CorpusID:237339921.

[98] Mnigming Guo, Niki Pissinou, and S.S. Iyengar. "Privacy-Preserving Deep Learning for Enabling Big Edge Data Analytics in Internet of Things". In: *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*. 2019, pp. 1–6.

[99] Jiawei Han, Micheline Kamber, and Jian Pei. "4 - Data Warehousing and Online Analytical Processing". In: *Data Mining (Third Edition)*. Ed. by Jiawei Han, Micheline Kamber, and Jian Pei. Third Edition. The Morgan Kaufmann Series in Data Management Systems. Boston: Morgan Kaufmann, 2012, pp. 125–185. ISBN: 978-0-12-381479-1. DOI: https://doi.org/10.1016/B978-0-12-381479-1.00004-6. URL: https://www.sciencedirect.com/science/article/pii/B9780123814791000046.

[100] Rebecca Hermon and Patricia A. H. Williams. "Big data in healthcare: What is it used for?" In: 2014. URL: https://api.semanticscholar.org/CorpusID:167190641.

[101] R. Himmer"oder et al. "On a Declarative Semantics for Web Queries". In: *In:* 5 (1997), pp. 386–398.

[102] Baik Hoh and Marco Gruteser. "Protecting Location Privacy Through Path Confusion". In: Oct. 2005, pp. 194–205. ISBN: 0-7695-2369-2. DOI: 10.1109/SECURECOMM.2005.33.

[103] Katsuhiro Honda et al. "A Collaborative Framework for Privacy Preserving Fuzzy Co-Clustering of Vertically Distributed Cooccurrence Matrices". In: *Advances in Fuzzy Systems* 2015 (2015), art. 729072.

[104] Belayet HossainDS. *Cancer*. 2023. URL: https://www.kaggle.com/datasets/belayethossainds/cancerand-deaths-dataset-19902019-globally.

[105] Ying Hu et al. "Simultaneously aided diagnosis model for outpatient departments via healthcare big data analytics". In: *Multim. Tools Appl.* 77.3 (2018), pp. 3729–3743.

[106] T. Hulsen et al. "From Big Data to Precision Medicine". In: *Frontiers in Medicine 6* (2019).

[107] Bill Inmon. *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*. 1st. Denville, NJ, USA: Technics Publications, LLC, 2016. ISBN: 1634621174.

[108] W.H. Inmon, Derek Strauss, and Genia Neushloss. "A brief history of data warehousing and first-generation data warehouses". In: Dec. 2008,

pp. 1–22. ISBN: 9780123743190. DOI: 10.1016/B978-0-12-374319-0.00001-4.

[109] Siriwan Intawichai and Saifon Chaturantabut. "Missing Image Data Reconstruction Based on Least-Squares Approach with Randomized SVD". In: Feb. 2021, pp. 1059–1070. ISBN: 978-3-030-68153-1. DOI: 10.1007/978-3-030-68154-8_89.

[110] Hewapathirana Ishara and Silva Thushari. "A big Data Analytics Framework for the Integration of Heterogeneous Federated Data Centers". In: Jan. 2021, pp. 650–657. DOI: 10.1109/ICICT50816.2021.9358503.

[111] Weizhao Jin et al. "FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System". In: *arXiv e-prints*, arXiv:2303.10837 (Mar. 2023), arXiv:2303.10837. DOI: 10.48550/arXiv.2303.10837. arXiv: 2303.10837 [cs.LG].

[112] Anish Jindal et al. "Providing Healthcare-as-a-Service Using Fuzzy Rule Based Big Data Analytics in Cloud Computing". In: *IEEE J. Biomed. Health Informatics* 22.5 (2018), pp. 1605–1618.

[113] Weipeng Jing et al. "Data Loss and Reconstruction of Location Differential Privacy Protection Based on Edge Computing". In: *IEEE Access* PP (June 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2922293.

[114] Yoon Jinsung et al. "EHR-Safe: generating high-fidelity and privacy-preserving synthetic electronic health records". In: *npj Digital Medicine* (2023).

[115] Tomcy John and Pankaj Misra. *Data Lake for Enterprises*. en. 1st ed. Birmingham, England: Packt Publishing, May 2017.

[116] Roberto J. Bayardo Jr. and Rakesh Agrawal. "Data Privacy through Optimal k-Anonymization". In: *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*. IEEE Computer Society, 2005, pp. 217–228.

[117] A Karr et al. "New Measures of Data Utility". In: *Workshop Manuscripts of Data Confidentiality, A Working Group in National Defense and Homeland Security*. 2006.

[118] Alan F Karr et al. "A Framework for Evaluating the Utility of Data Altered to Protect Confidentiality". In: *The American Statistician* 60.3 (2006), pp. 224–232.

[119] Marwa Keshk et al. "Privacy-Preserving Big Data Analytics for Cyber-Physical Systems". In: *Wirel. Netw.* 28.3 (Apr. 2022), pp. 1241–1249.

[120] Çetin Kaya Koç, Funda Özdemir, and Zeynep Ödemiş Özger. "Boneh-Goh-Nissim Algorithm". In: *Partially Homomorphic Encryption*. Cham: Springer International Publishing, 2021, pp. 123–133. ISBN: 978-3-030-87629-6. DOI: 10.1007/978-3-030-87629-6_11.

[121] Julia Kokina, Dessislava Pachamanova, and Andrew Corbett. "The role of data visualization and analytics in performance management: Guiding entrepreneurial growth decisions". In: *Journal of Accounting Education* 38 (Jan. 2017). DOI: 10.1016/j.jaccedu.2016.12.005.

[122] Helena Kościelniak and Agnieszka Puto. "BIG DATA in Decision Making Processes of Enterprises". In: *Procedia Computer Science* 65 (2015). International Conference on Communications, management, and Information technology (ICCMIT'2015), pp. 1052–1058. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.09.053.

[123] M. R. Kosorok and E. B. Laber. "Precision Medicine". In: *Annual Review of Statistics and its Application 6* (2019), pp. 263–286.

[124] Sinan Kufeoglu. "Emerging Technologies". In: July 2022, pp. 41–190. ISBN: 978-3-031-07126-3. DOI: 10.1007/978-3-031-07127-0_2.

[125] Bai Li et al. "A Privacy Preserving Algorithm to Release Sparse High-dimensional Histograms". In: *Journal of Privacy and Confidentiality* 8 (Dec. 2018). DOI: 10.29012/jpc.657.

[126] Jiuyong Li et al. "Achieving k-Anonymity by Clustering in Attribute Hierarchical Structures". In: *Data Warehousing and Knowledge Discovery, 8th International Conference, DaWaK 2006, Krakow, Poland, September 4-8, 2006, Proceedings*. Vol. 4081. Lecture Notes in Computer Science. Springer, 2006, pp. 405–416.

[127] Jiuyong Li et al. "Anonymization by Local Recoding in Data with Attribute Hierarchical Taxonomies". In: *Knowledge and Data Engineering, IEEE Transactions on* 20 (Oct. 2008), pp. 1181–1194. DOI: 10.1109/TKDE.2008.52.

[128] Canada Open Government Licence. *NARCAN ADMINISTRATION*. 2021. URL: https://data.winnipeg.ca/Fireand-Paramedic-Service/Narcan-Administrations/qd6b-q49i.

[129] Canada Open Government Licence. *SUBSTANCE USE*. 2021. URL: https://data.winnipeg.ca/Fire-andParamedic-Service/Substance-Use/6x82-bz5y.

[130] Chi Lin et al. "Differential Privacy Preserving in Big Data Analytics for Connected Health". In: *Journal of Medical Systems* 40 (Feb. 2016). DOI: 10.1007/s10916-016-0446-0.

[131] Huadong Liu et al. "OPERA: Optional Dimensional Privacy-Preserving Data Aggregation for Smart Healthcare Systems". In: *IEEE Transactions on Industrial Informatics* 19.1 (2023), pp. 857–866. DOI: 10.1109/TII.2022.3192037.

[132] Z. Liu and A. Zhang. "Sampling for Big Data Profiling: A Survey". In: *IEEE Access 8* (2020), pp. 72713–72726.

[133] Zhusen Liu et al. "EPMDA-FED: Efficient and Privacy-Preserving Multidimensional Data Aggregation Scheme With Fast Error Detection in Smart Grid". In: *IEEE Internet of Things Journal* 9.9 (2022), pp. 6922–6933. DOI: 10.1109/JIOT.2021.3113519.

[134] Ashwin Machanavajjhala et al. "L-Diversity: Privacy beyond k-Anonymity". In: 1.1 (Mar. 2007), 3–es. ISSN: 1556-4681. DOI: 10.1145/1217299.1217302. URL: https://doi.org/10.1145/1217299.1217302.

[135] Suresh Madhav et al. "KloakDB: A Platform for Analyzing Sensitive Data with K-anonymous Query Processing". In: (Mar. 2019).

[136] Abdul Majeed. "Attribute-centric anonymization scheme for improving user privacy and utility of publishing e-health data". In: *Journal of King Saud University - Computer and Information Sciences* 31.4 (2019), pp. 426–435. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2018.03.014.

[137] Brijesh B. Mehta and Udai Pratap Rao. "Improved l-diversity: Scalable anonymization approach for Privacy Preserving Big Data Publishing". In: *Journal of King Saud University - Computer and Information Sciences* 34.4 (2022), pp. 1423–1430. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2019.08.006.

[138] M. Milani, Y. Huang, and F. Chiang. "Preserving Diversity in Anonymized Data". In: 24th International Conference on Extending Database Technology. 2021, pp. 511–516.

[139] M. J. Milenkovic, A. Vukmirovic, and D. Milenkovic. "Big Data Analytics in the Health Sector: Challenges and Potentials". In: *Management: Journal of Sustainable Business and Management Solutions in Emerging Economies 24(1)* (2019), pp. 23–33.

[140] J. Minou et al. "Health Professionals' Perception about Big Data Technology in Greece". In: *Acta Informatica Medica 28(1)* (2020), pp. 48–51.

[141] Noman Mohammed et al. "Centralized and Distributed Anonymization for High-Dimensional Healthcare Data". In: *ACM Trans. Knowl. Discov. Data* 4.4 (Oct. 2010). ISSN: 1556-4681. DOI: 10.1145/1857947.1857950. URL: https://doi.org/10.1145/1857947.1857950.

[142] Ahsan Mominul, Tea S., and Albarbar A. "Development of Novel Big Data Analytics Framework for Smart Clothing". In: *IEEE Access* PP (Aug. 2020), pp. 1–1. DOI: 10.1109/ACCESS.2020.3015152.

[143] "Mondrian Multidimensional K-Anonymity". In: (2006), pp. 25–25. DOI: 10.1109/ICDE.2006.101.

[144] Sara Mumtaz, Azhar Rauf, and Shah Khusro. "A distortion based technique for preserving privacy in OLAP data cube". In: *International Conference on Computer Networks and Information Technology*. 2011, pp. 185–189. DOI: 10.1109/ICCNIT.2011.6020928.

[145] A. Muniasamy et al. "Deep Learning for Predictive Analytics in Healthcare". In: International Conference on Advanced Machine Learning Technologies and Applications. 2019, pp. 32–42.

[146] Krishnamurty Muralidhar, Rahul Parsa, and Rathindra Sarathy. "A General Additive Data Perturbation Method for Database Security". In: *Management Science* 45.10 (1999), pp. 1399–1415. ISSN: 00251909, 15265501. URL: http://www.jstor.org/stable/2634846 (visited on 11/13/2022).

[147] A. Nadeem and M. Y. Javed. "A Performance Comparison of Data Encryption Algorithms". In: IEEE International Conference on Information and Communication Technologies. 2005, pp. 84–89.

[148] S. J. Nass, L. A. Levit, and L. O. Gostin. *Beyond the HIPAA Privacy Rule*. USA: National Academies Press, 2009.

[149] Mehmet Ercan Nergiz and Chris Clifton. "Thoughts on k-Anonymization". In: *Data & Knowledge Engineering* 63.3 (2007), pp. 622–645.

[150] T. Neubauer and J. Heurix. "A Methodology for the Pseudonymization of Medical Data". In: *International Journal of Medical Informatics 80(3)* (2011), pp. 190–204.

[151] Axel Oehmichen et al. "OPAL: High performance platform for large-scale privacy-preserving location data analytics". In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 1332–1342.

[152] Andrew Onesimu, J. Karthikeyan, and Yuichi Sei. "An efficient clustering-based anonymization scheme for privacy-preserving data collection in IoT based healthcare services". In: *Peer-to-Peer Networking and Applications* 14 (Feb. 2021). DOI: 10.1007/s12083-021-01077-7.

[153] Andrew Onesimu et al. "Privacy Preserving Attribute-Focused Anonymization Scheme for Healthcare Data Publishing". In: *IEEE Access* PP (Jan. 2022), pp. 1–1. DOI: 10.1109/ACCESS.2022.3199433.

[154] OpenHMS. *RMIIO*. https://openhms.sourceforge.io/rmiio/. 2007.

[155] Soufiene Othman et al. "Privacy-preserving aware data aggregation for IoT-based healthcare with green computing technologies". In: *Computers and Electrical Engineering* 101 (Apr. 2022). DOI: 10.1016/j.compeleceng.2022.108025.

[156] Hae-Sang Park and Chi-Hyuck Jun. "A Simple and Fast Algorithm for K-Medoids Clustering". In: *Expert Syst. Appl.* 36.2 (Mar. 2009), pp. 3336–3341. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2008.01.039. URL: https://doi.org/10.1016/j.eswa.2008.01.039.

[157] R. Pastorino et al. "Benefits and Challenges of Big Data in Healthcare: An Overview of the European Initiatives". In: *European Journal of Public Health 29(3)* (2019), pp. 23–27.

[158] Cong Peng et al. "An Efficient Privacy-Preserving Aggregation Scheme for Multidimensional Data in IoT". In: *IEEE Internet of Things Journal* 9.1 (2022), pp. 589–600. DOI: 10.1109/JIOT.2021.3083136.

[159] Rishabh Poddar et al. "Visor: Privacy-Preserving Video Analytics as a Cloud Service". In: *ArXiv* abs/2006.09628 (2020). URL: https://api.semanticscholar.org/CorpusID:219721137.

[160] Fabian Prasser et al. "Lightning: Utility-Driven Anonymization of High-Dimensional Data". In: *Trans. Data Privacy* 9.2 (Aug. 2016), pp. 161–185. ISSN: 1888-5063.

[161] Zhenquan Qin et al. "Nonlinear Traffic Data Reconstruction in large-scale Internet of Vehicle Systems: A neural network approach". In: *IEEE Access* PP (2022), pp. 1–1. URL: https://api.semanticscholar.org/CorpusID:247540096.

[162] Do Le Quoc et al. "Privacy-Preserving Data Analytics". English. In: *Encyclopedia of Big Data Technologies*. Ed. by Sherif Sakr and Al-

bert Zomaya. 1st ed. Springer-Verlag, Mar. 2019, pp. 1292–1300. ISBN: 3319775243. DOI: 10.1007/978-3-319-77525-8_153.

[163] Maria Rashidi et al. "Decision Support Systems". In: Oct. 2018, pp. 19–38. ISBN: 978-1-78984-197-8. DOI: 10.5772/intechopen.79390.

[164] Franck Ravat and Yan Zhao. "Data Lakes: Trends and Perspectives". In: *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part I*. Lecture Notes in Computer Science. Springer, 2019, pp. 304–313.

[165] Imran Razzak, Peter W. Eklund, and Guandong Xu. "Improving healthcare outcomes using multimedia big data analytics". In: *Neural Comput. Appl.* 34.17 (2022), pp. 15095–15097.

[166] National Cancer Registration and Analysis Service. *Health Data Insight: The Simulacrum*. Available at: . URL: https://www.cancerdata.nhs.uk/simulacrum.

[167] Wang Ren et al. "Privacy Enhancing Techniques in the Internet of Things Using Data Anonymisation". In: *Information Systems Frontiers* (May 2021). DOI: 10.1007/s10796-021-10116-w.

[168] SAS Reports. *Cebr: Data equity, Unlocking the value of big data.* https://www.sas.com/content/dam/SAS/en_gb/doc/analystreport/cebr-value-of-big-data.pdf. pp. 1-44. 2012.

[169] John Resig. *JQuery*. https://jquery.com/. 2006.

[170] Deepjyoti Roy and Mala Dutta. "A systematic review and research perspective on recommender systems". In: *Journal of Big Data* 9.1 (May 2022), p. 59. ISSN: 2196-1115. DOI: 10.1186/s40537-022-00592-5. URL: https://doi.org/10.1186/s40537-022-00592-5.

[171] Tharuka Rupasinghe, Frada Burstein, and Carsten Rudolph. "Blockchain based dynamic patient consent: a privacy-preserving data acquisition architecture for clinical data analytics". In: 2019.

[172] Philip Russom. "Big data analytics". In: *TDWI best practices report, fourth quarter* 19.4 (2011), pp. 1–34.

[173] Firas Saidi, Zouheir Trabelsi, and Henda Ben Ghezala. "Towards a Multidimensional Model for Terrorist Attacks Analysis and Mining". In: *2018 28th International Conference on Computer Theory and Applications (ICCTA)*. 2018, pp. 55–59. DOI: 10.1109/ICCTA45985.2018.9499167.

[174] Sinha Sapna, Bhatnagar Vishal, and Bansal Abhay. "A Framework for Effective Data Analytics for Tourism Sector: Big Data Approach". In: *International Journal of Grid and High Performance Computing* 9 (Oct. 2017), pp. 92–104. DOI: 10.4018/IJGHPC.2017100106.

[175] Philip Sedgwick. "Pearson's correlation coefficient". In: *BMJ* 345 (July 2012), e4483–e4483. DOI: 10.1136/bmj.e4483.

[176] Philip Sedgwick. "Spearman's rank correlation coefficient". In: *BMJ: British Medical Journal* 349 (Nov. 2014), g7327. DOI: 10.1136/bmj.g7327.

[177] E. Serra et al. "An effective and efficient graph representation learning approach for big graphs". In: *SEBD* (2021). URL: http://ceur-ws.org/Vol-2994/paper13.pdf.

[178] S. Shafqat et al. "Leveraging Deep Learning for Designing Healthcare Analytics Heuristic for Diagnostics". In: *Neural Processing Letters* (2021), pp. 1–27.

[179] Jawwad A Shamsi and Muhammad Ali Khojaye. "Understanding privacy violations in big data systems". In: *It Professional* 20.3 (2018), pp. 73–81.

[180] He Shan et al. "Integrating a Federated Healthcare Data Query Platform With Electronic IRB Information Systems". In: *Annual Symposium proceedings / AMIA Symposium. AMIA Symposium* 2010 (Nov. 2010), pp. 291–5.

[181] Amit P. Sheth and James A. Larson. "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases". In: *ACM Computing Surveys* 22.3 (1990), pp. 183–236.

[182] Juanjuan Shi. "Music Recommendation Algorithm Based on Multidimensional Time-Series Model Analysis". In: *Complex.* 2021 (2021), 5579086:1–5579086:11. DOI: 10.1155/2021/5579086. URL: https://doi.org/10.1155/2021/5579086.

[183] D. Singh and C. K. Reddy. "A Survey on Platforms for Big Data Analytics". In: *Journal of Big Data 2(1)* (2015), pp. 1–20.

[184] R. K. Som. *Practical Sampling Techniques*. USA: CRC Press, 1995.

[185] Song. *Data Warehousing Systems: Foundations and Architectures*. Jan. 2016, pp. 1–11. ISBN: 978-1-4899-7993-3. DOI: 10.1007/978-1-4899-7993-3_121-3.

[186] Rafael Staib. *Jquery Steps*. http://www.jquery-steps.com/. 2013.

[187] R Sudha et al. "Enhanced Data Privacy Using Vertical Fragmentation and Data Anonymization Techniques". In: Dec. 2021. ISBN: 9781643682167. DOI: 10.3233/APC210292.

[188] L. Sweeney. "K-anonymity. A Model for Protecting Privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5)* (2002), pp. 557–570.

[189] Latanya Sweeney. "K-Anonymity: A Model for Protecting Privacy". In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10.5 (Oct. 2002), pp. 557–570. ISSN: 0218-4885. DOI: 10.1142/S0218488502001648. URL: https://doi.org/10.1142/S0218488502001648.

[190] Matilde Tanaglia et al. "Multidimensional Design and Analysis of a Data Mart Related to Healthcare Treatments with Biologic Drugs". In: Sept. 2021, pp. 1–7. DOI: 10.1109/ISCC53001.2021.9631489.

[191] J. Teng. *SEER Breast Cancer Data*. Available at: s. URL: https://ieee-dataport.org/open-access/seer-breast-cancer-data.

[192] Rick F. van der Lans. "Chapter 1 - Introduction to Data Virtualization". In: *Data Virtualization for Business Intelligence Systems*. Boston: Morgan Kaufmann, 2012, pp. 1–26.

[193] Prasad Velpula and Rajendra Pamula. "CEECP: CT-based enhanced e-clinical pathways in terms of processing time to enable big data analytics in healthcare along with cloud computing". In: *Comput. Ind. Eng.* 168 (2022), p. 108037.

[194] Charles Vesteghem et al. "Implementing the FAIR Data Principles in Precision Oncology: Review of Supporting Initiatives". In: *Briefings in Bioinformatics* 21.3 (2020), pp. 936–945.

[195] Marco Viceconti, Peter Hunter, and Rod Hose. "Big Data, Big Knowledge: Big Data for Personalized Healthcare". In: *IEEE Journal of Biomedical and Health Informatics* 19.4 (2015), pp. 1209–1215. DOI: 10.1109/JBHI.2015.2406883.

[196] J. S. Vitter. "Random Sampling with a Reservoir". In: *ACM Transactions on Mathematical Software 11(1)* (1985), pp. 37–57.

[197] Jinyan Wang et al. "Two privacy-preserving approaches for data publishing with identity reservation". In: *Knowledge and Information Systems* 60 (Aug. 2019). DOI: 10.1007/s10115-018-1237-3.

[198] L. Wang et al. "pipsCloud: High Performance Cloud Computing for Remote Sensing Big Data Management and Processing". In: *Future Generation Computer Systems 78* (2018), pp. 353–368.

[199] Yichuan Wang and Nick Hajli. "Exploring the path to big data analytics success in healthcare". In: *Journal of Business Research* 70 (2017), pp. 287–299. ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2016.08.002.

[200] Griffin M. Weber et al. "The Shared Health Research Information Network (SHRINE): A Prototype Federated Query Tool for Clinical Data Repositories". In: *Journal of the American Medical Informatics Association* 16.5 (2009), pp. 624–630. ISSN: 1067-5027. DOI: https://doi.org/10.1197/jamia.M3191. URL: https://www.sciencedirect.com/science/article/pii/S1067502709001327.

[201] WHO. *Cancer Incidence*. 2023. URL: https://ci5.iarc.fr/CI5plus/Default.aspx.

[202] Katja Wikström et al. "Electronic Health Records as Valuable Data Sources in the Health Care Quality Improvement Process". In: *Health Services Research and Managerial Epidemiology* 6 (May 2019), p. 233339281985287. DOI: 10.1177/2333392819852879.

[203] Mark Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3 (Mar. 2016). DOI: 10.1038/sdata.2016.18.

[204] Guangjun Wu et al. "Privacy-Preserved Electronic Medical Record Exchanging and Sharing: A Blockchain-Based Smart Healthcare System". In: *IEEE Journal of Biomedical and Health Informatics* 26 (2021), pp. 1917–1927. URL: https://api.semanticscholar.org/CorpusID:240229335.

[205] D. Xiang and W. Cai. "Privacy Protection and Secondary Use of Health Data: Strategies and Methods". In: *BioMed Research International* 6967 (2021).

[206] Chugui Xu et al. "DPPro: Differentially Private High-Dimensional Data Release via Random Projection". In: *IEEE Transactions on Information Forensics and Security* 12.12 (2017), pp. 3081–3093. DOI: 10.1109/TIFS.2017.2737966.

[207] Jian Xu et al. "Utility-based anonymization for privacy preservation with less information loss". In: *SIGKDD Explorations* 8 (Dec. 2006), pp. 21–30. DOI: 10.1145/1233321.1233324.

[208] Jian Xu et al. "Utility-based anonymization using local recoding". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, pp. 785–790.

[209] Min Xu et al. "DPSAaS: multi-dimensional data sharing and analytics as services under local differential privacy". In: *Proceedings of the VLDB Endowment* 12 (Aug. 2019), pp. 1862–1865. DOI: 10.14778/3352063.3352085.

[210] Chaowei Yang et al. "Big Data and cloud computing: innovation opportunities and challenges". In: *Int. J. Digit. Earth* 10.1 (2017), pp. 13–53.

[211] Matei Zaharia et al. "Apache Spark: A Unified Engine for Big Data Processing". In: *Commun. ACM* 59.11 (2016), pp. 56–65.

[212] Matei A. Zaharia et al. "Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics". In: *Conference on Innovative Data Systems Research*. 2021. URL: https://api.semanticscholar.org/CorpusID:229576171.

[213] Hessam Zakerzadeh, Charu C. Aggrawal, and Ken Barker. *Towards Breaking the Curse of Dimensionality for High-Dimensional Privacy: An Extended Version*. 2014. arXiv: 1401.1174 [cs.DB].

[214] Kirtirajsinh Zala et al. "PRMS: Design and Development of Patients' E-Healthcare Records Management System for Privacy Preservation in Third Party Cloud Platforms". In: *IEEE Access* 10 (Jan. 2022), pp. 1–1. DOI: 10.1109/ACCESS.2022.3198094.

[215] Jiale Zhang et al. "PEFL: A Privacy-Enhanced Federated Learning Scheme for Big Data Analytics". In: Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9014272.

[216] Qiang Zhang et al. "Missing Data Reconstruction in Remote Sensing Image With a Unified Spatial-Temporal-Spectral Deep Convolutional Neural Network". In: *IEEE Transactions on Geoscience and Remote Sensing* PP (Feb. 2018). DOI: 10.1109/TGRS.2018.2810208.

[217] Xiaojun Zhang et al. "Privacy-preserving statistical analysis over multi-dimensional aggregated data in edge computing-based smart grid systems". In: *Journal of Systems Architecture* 127 (2022), p. 102508. ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2022.102508.

[218] Xuyun Zhang et al. “MRMondrian: Scalable Multidimensional Anonymisation for Big Data Privacy Preservation”. In: *IEEE Transactions on Big Data* PP (Dec. 2017), pp. 1–1. DOI: 10.1109/TBDATA.2017.2787661.

[219] Yandong Zheng et al. “PMRQ: Achieving Efficient and Privacy-Preserving Multi-Dimensional Range Query in eHealthcare”. In: *IEEE Internet of Things Journal* 9 (Sept. 2022), pp. 1–1. DOI: 10.1109/JIOT.2022.3158321.

8

Other Publications

[65] Alfredo Cuzzocrea et al. "F-TBDA: A Frequency-Based Temporal Big Data Analytics Technique for Mining and Analyzing Quality-Of-Life Indicators of Cancer Patients". In: *2023 IEEE International Conference on Big Data (BigData)*. 2023, pp. 5197–5205. DOI: 10.1109/BigData59044.2023.10386767.

[66] Alfredo Cuzzocrea et al. "OLAP over Big COVID-19 Data: A Real-Life Case Study". In: *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. 2022, pp. 1–6. DOI: 10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9927803.

[68] Alfredo Cuzzocrea et al. "Scaling Posterior Distributions over Differently-Curated Datasets: A Bayesian-Neural-Networks Methodology". In: *Foundations of Intelligent Systems*. Cham: Springer International Publishing, 2022, pp. 198–208. ISBN: 978-3-031-16564-1.

[70] Alfredo Cuzzocrea et al. "The Emerging Challenges of Big Data Lakes, and a Real-Life Framework for Representing, Managing and Supporting Machine Learning on Big Arctic Data". In: *Advances in Intelligent Networking and Collaborative Systems*. Ed. by Leonard Barolli and Hiroyoshi Miwa. Cham: Springer International Publishing, 2022, pp. 161–174. ISBN: 978-3-031-14627-5.

[177] E. Serra et al. "An effective and efficient graph representation learning approach for big graphs". In: *SEBD* (2021). URL: http://ceur-ws.org/Vol-2994/paper13.pdf.