Leveraging profiling to bridge healthcare silos for federated analyses

Nelly Barret[1 [0000−0002−3469−4149]], Anna Bernasconi[1 [0000−0001−8016−5750]], Cinzia Cappiello[1 [0000−0001−6062−5174]], Giacomo Palu[1], and Pietro Pinoli[1 [0000−0001−9786−2851]]

Politecnico di Milano, Italy {firstname.lastname}@polimi.it

Abstract. Healthcare is more and more relying on digital information, bringing new challenges for its management, exploration, and usage. Healthcare data represents a challenge for information systems because, for privacy regulations, it cannot exit the original silo in which it has been produced (typically owned by hospitals), and may be of various kinds (clinical reports, DNA sequences, MRI scans, etc). To manage this complexity, it is natural to use Federated Learning to safely analyze the underlying silos' content. However, designing and running federated algorithms requires to know what the silos contain and how they can be joined (on which common attributes). Existing catalogs provide preliminary visualizations, which are hardly generalizable due to their underlying use-case-tailored data models. To overcome these limitations, we provide a general catalog conceptual model as well as profiling techniques to extract information of interest from silos. Our proposed catalog is general enough to be used in various healthcare scenarios with diverse kinds of data. It also facilitates experts' work in creating Federated Learning algorithms running in networks of interoperable healthcare silos.

Keywords: Conceptual model · Healthcare data · Federated learning.

1 Introduction

The world's digitalization has led to unprecedented data creation rates in various industries, such as healthcare, transportation, social media, and education. To handle massive production and sharing, data often resides in more or less curated and interoperable silos maintained by data owners. Domain experts usually employ several silos simultaneously, e.g, to obtain more and/or finer information, whether this is with queries or federated learning (FL) algorithms. As an example, consider medical practitioners working on cancer: they seek to answer questions, such as "Which genes favor severe COVID-19 forms for kidney cancer patients?". For this task, they first need to identify silos of interest by inspecting stored data, then proceed with the design of federated analyses on the chosen datasets. Such course of actions allows to (*i*) combine several datasets (clinical reports, DNA analysis, scans, etc) in a given silo; and (*ii*) enrich existing data in a silo with data present in other silos in terms of number of patients and/or

number of features. In real-world scenarios, combining several datasets originating from a silo is a complex task; extending them with datasets from other silos is an even harder task. First, silos are created independently by actors, thus are not interoperable due to their high heterogeneity (inside and across silos). Second, they may contain sensitive data, which prevents the creation of a single curated silo according to current regulations, e.g., the GDPR. Third, users need to know which data is available in the silos and how it relates to other silos in order to formulate coherent federated analyses. These three reasons hinder the federated analyses of healthcare silos.

Toward bridging silos together and enabling cooperation between hospitals, we have initiated **I-ETL** [1], a framework to build healthcare networks with interoperability as a first-class citizen. In that work, we introduce a pipeline to make data (more) interoperable according to the metadata specified by experts as well as two novel general healthcare conceptual models for metadata and data. However, the I-ETL network cannot be used as such because, before formulating federated analyses, metadata and data of each silo require to be first discovered by means of a catalog. Therefore, we propose our twofold vision in this paper: (*i*) a **conceptual model** for cataloging silos' metadata and data in the I-ETL network; and (*ii*) a **pipeline to build a general catalog** implementing our conceptual model together with profiling techniques. Our paper is organized as follows. We first motivate our work with a real healthcare FL scenario (Sec. 2). Then, we present our approach (Sec. 3). We finally discuss related work (Sec. 4) before concluding (Sec. 5).

2 Motivating example: FL for cancer and COVID-19

We start with a motivating example featuring open data collected for ESKD (end-stage kidney disease) patients having COVID-19 [9, 7]. ESKD is the last stage of chronic kidney disease, leading to slow kidney functioning and higher risks of severe COVID-19, thus necessitating extra care. In our example, we consider two hospitals. The former contains patient phenotypic data (age, ethnicity, life habits, etc.) and whether/how they were affected by COVID-19 (severity, nasal tests results, and MRI scans of their lungs). The latter contains genomic data collected during their ESKD analysis, i.e., RNA sequence counts for a panel of 60k genes. Having different kinds of data distributed across hospitals aligns with the typical situation where only a few large hospitals can run genetic analyses of their patients' DNA, due to the high monetary cost. In our scenario, experts seek to answer two questions: (*i*) "*How are COVID-19 symptoms amplified for ESKD patients?*"; and (*ii*) "*Which ESKD-related genes favor severe COVID-19 forms?*". Keeping them in mind, we explain three tasks that healthcare experts will experience.

Task 1: explore silos through the catalog. Toward answering the above questions, healthcare experts first explore the datasets available in the two hospitals by browsing the general catalog. During this task, they can find out that 80% of patients are aged above 60yo, while the remaining ones are scattered between 50-59yo (10 patients) and 20-49yo (same). They can also see that males

are prevalent (80%). Those observations will help them in interpreting the results of FL analyses and AI algorithms.

Task 2: run a FL analysis. A second task is to define a simple federated learning analysis, e.g., to compute the distribution of patients with severe COVID-19 symptoms based on the number of comorbidities they may have. This analysis requires to join the two silos based on patient identifiers, a task facilitated by the high interoperability in the network, guaranteed by applying I-ETL.

Task 3: train a federated AI algorithm. A final task is to formulate complex federated AI algorithms, e.g., to predict whether a new ESKD patient will develop severe COVID-19 forms based on its phenotypic, biological, and genetic data. This can be done by training a binary classifier (decision trees, logistic regression, etc.) in a federated manner: individual models are trained locally in each silo and are subsequently aggregated in a general binary classifier. Afterwards, the model can be used for prediction or re-trained with different parameters and/or new data.

3 Modeling and profiling healthcare silos

Figure 1 illustrates our approach, starting from the heterogeneous datasets residing in different hospitals' silos to the construction of the general catalog for exploring them and running federated analyses. Starting from the left, hospitals H_i have datasets $d_i^1..d_i^n$ to explore and use for further studies. Next, I-ETL [1] is run at each hospital on the set of datasets and produces an interoperable silo S_i containing all such data. Subsequently, we **profile** (Section 3.2) each silo to obtain an **individual catalog** C_i, an instance of our catalog conceptual model (Section 3.1). The set of individual catalogs is then merged into **a single general catalog**. Lastly, experts can inspect the silos' content through the Web interface and design **federated learning (FL) analyses and algorithms**. To provide a secure and efficient architecture for that, we rely on the PHT [2] (Personal Health Train) and PADME [12]. Introduced in 2020, the **PHT** is a novel approach to enable FL between institutions that cannot centralize nor share their data without privacy risks. It follows a decentralized scheme where data always remains in the original silo, named a *station*. Next, FL tasks are encapsulated into *trains*, which collect intermediary results by going through all the stations. The train ends its road in the *central station* which takes care of aggregating the intermediate results and returns the final results to the expert. Released in 2022, **PADME** is an implementation of the PHT, widely adopted for its generality and applicability to several settings, including healthcare.

3.1 The catalog conceptual model

We seek a general catalog, flexible to various kinds of data and adapted to our silos. In turn, our conceptual model (*i*) builds on the I-ETL data model, promoting features (variable) and records (value for a given variable) for generality; (*ii*) profiles data (aggregates) and domain metadata (knowledge provided by experts about silos' data); (*iii*) allows for generic instantiation of charts for all

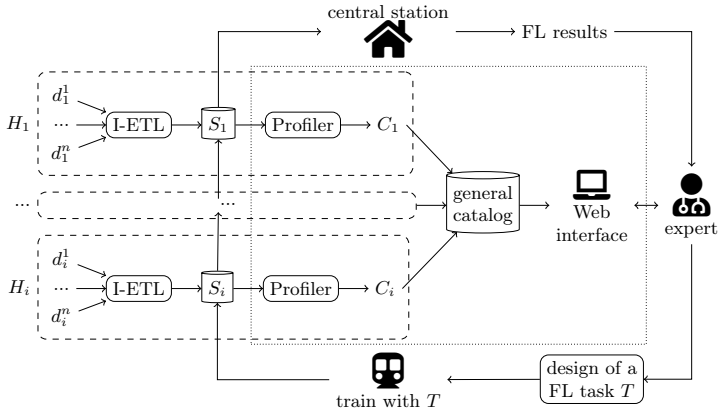

Fig. 1. Overview of our approach with processes (rounded boxes), data storage (cylinders), sequence flow (arrows). Dashed lines delimit hospitals' boundaries; dotted ones delimit our novel contributions.

features; and (*iv*) provides useful information to PADME to run FL tasks such as information on stations' capacities and data statistics. Figure 2 presents our catalog conceptual model. Rectangles are entities, rounded boxes are relationships and triangles are specializations. Primary keys are in **bold** and mandatory attributes are underlined. Our cardinalities adopt the notation of [4], e.g., a Dataset is composed of 1 to n Features, while it has exactly one DatasetProfile.

The central entity is the DATASET. Each dataset has an identifier *Did*, a *version* number (such as "2.1"), and an initial *releaseDate*. It may also have a *lastUpdate*, a text describing the last version's changes (*versionNotes*), and a *license* under which it can be used. The current version and the initial release date of each dataset are mandatory. Datasets are then analyzed in order to obtain a DATASETPROFILE composed of: a unique identifier *DPid*, a short textual *description*, a *theme* (e.g., covid), the dataset's *fileType*s (csv, xlsx, vcf, etc.), the dataset *size* in Mb, and the number of tuples (or the number of files if the data is not tabular) in *nbTuples*. The tuple completeness *tpCompl* is the ratio of patients with at least one record per feature while the tuple uniqueness *tpUniqueness* is the ratio of patients with no more than one record per feature.

Each dataset is composed of a set of FEATURES (variables), each defined by an identifier *Fid* and a *name* (both mandatory). Further, it may have a short text for *description*, an *ontology* to identify the represented concept in existing specialized ontologies (like SNOMED-CT, LOINC, or OrphaNet), and a *dataType* to specify the expected value type (integer, boolean, numeric, string, etc.). Next, the attribute *visibility* specifies whether data is shown with or without anonymization, e.g., dates are partially anonymized by removing the day while numeric values are not anonymized. The *entityType* is the type of data the feature represents (clinical, genomic, imaging, etc.). In our motivating example, feature examples are: patient age, ethnicity, smoking habits, visual artifacts in MRI lung scans, various DNA sequence counts, etc. The patient age feature would be instantiated

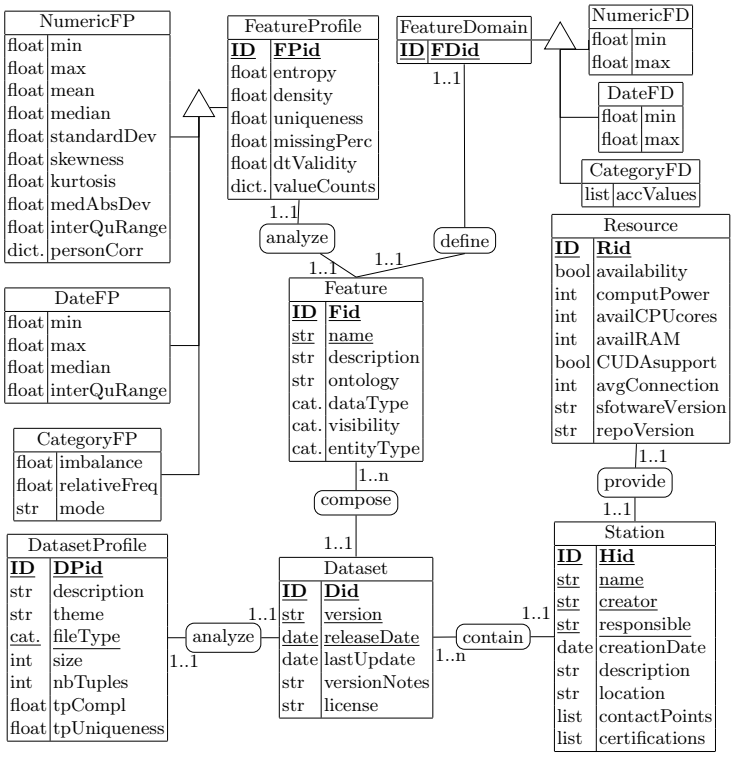

Fig. 2. The conceptual model we propose for individual and general catalogs.

with *description*="Patient age in years", *ontology*=https://loinc.org/30525-0, *dataType*="integer", *visibility*="not_anonymized", and *entityType*="phenotypic".

Further, each feature is defined on a FeatureDomain, which is refined based on the feature *dataType*. For categorical features, their domain is limited to a list of values (*accValues* in CategoryFD). NumericFD and DateFD limit values using *min* and *max* values. For instance, our patient age feature has a numeric feature domain, whose min and max values are 0 and 120.

Next, the feature is analyzed to obtain its FeatureProfile. It holds 6 attributes, 5 for statistics and the latter for aggregated data. Precisely, statistics are: the *entropy* (disorder within the feature's values), the *density* (distance from the uniform distribution), the *uniqueness* (percentage of distinct values), the *missingPerc* (percentage of null values), the *dtValidity* (ratio of values conforming to the feature's *dataType*). In our motivating example, the patient age feature would have a feature profile with the following values: *entropy*=0.775 (most values range between 60 and 90, but there are several younger patients, seen as "outliers"), *density*=0.01 (the maximum frequency is 7, thus there are many different values with a low frequency), *uniqueness*=0.39 (44 distinct values over 111 patients), *missingPerc*=0 (every patient has an age in the dataset), *dtValidity*=1 (all values could be cast to integers). Last, *valueCounts* associates each value with its frequency. This corresponds to the aggregated data that will be shown in the catalog; no individual data tuples can be shown.

Each feature's profile can be refined based on its type: numeric, date, or category, leading to the entities NumericFP, DateFP, and CategoryFP. Their attributes are computed on non-null values, except if otherwise specified, and are exemplified on the feature age. For numeric feature profiles, a set of 10 statistics are computed, including the *skewness* (distribution asymmetry; -0.95 indicates a right-placed distribution, attested by the old age range), the *kurtosis* (distribution "tailness"; 1.22 indicates few outliers – recall the presence of few younger patients), the *medAbsDev* (outlier-robust version of the mean; 9.41 means that most patients are $\pm$ 9.4yo around the range 70-80yo), the *interQuRange* (difference between the 3^{rd} and 1^{st} quartiles; 14yo is the maximum age difference for half of patients). Finally, the attribute *pearsonCorr* stores the Pearson correlation coefficients for that feature with the 10 features having the highest number of values. This significantly lowers the heavy computation of Pearson coefficients, which may not be tractable if there are more than a few dozen of features (which is usually the case for healthcare datasets). For instance, the Pearson value of disease fatality with the age and the smoking habits is only 0.114 and -0.167, meaning that a more subtle and complex correlation exists. The date feature profile entity leverages a subset of the numeric FP entity's attributes. Last, the category feature profile exhibits the *imbalance* (ratio of the highest and lowest value frequencies), the *relativeFreq* (ratio of the highest value frequency and the number of values), and the *mode* (the most frequent value). All such statistics are precious to experts for designing FL tasks and interpreting results, but also to the FL algorithms themselves.

Finally, the Station entity corresponds to a hospital and contains one or several datasets. It has a *name*, a *creator*, and a *responsible* (at a minimum). It may also have a *creationDate*, a short textual *description*, a *location*, a list of *contactPoints*, and a list of *certification* documents. Each station provides a computational Resource environment, exhibiting its current state: its *availability*, the available computational power (*computPower*), the number of available CPU cores (*availCPUcores*), the available RAM (*availRAM*), whether it supports CUDA for GPU computations (*CUDAsupport*), the average speed connection (*avgConnection*), the currently deployed software version (*softwareVersion*; PADME version here, but this can be adapted to the project).

3.2 Profiling silos for individual catalogs

Leveraging the conceptual model described in Section 3.1, we explain how to individually profile each silo in the network, leading to individual catalogs (C_1 to C_i in Figure 1). Our profiler tool works as follows:

1. Identify the Datasets in the silo;
2. Compute each DatasetProfile by gathering information previously provided by experts (including *description* and *theme* in Figure 2) as well as statistics computed from the dataset itself (such as its *size*).
3. Enumerate all the Features composing each Dataset and extract their metadata (*name*, *description*, *ontology*, etc.);
4. Extract the FeatureDomain, an information provided by experts in the metadata and based on the feature data type. Integer and float, respectively

date and datetime, features lead to NumericFD, respectively DateFD; boolean, string and category features lead to CategoryFD.
5. Analyze each Feature to obtain the according FeatureProfile. Statistics (*entropy*, *density*, etc.) and aggregated data (*valueCounts* in Figure 2) are computed directly in the silo using dedicated queries. The specializations are also made on the feature's data type.
6. Collect Hospital information about their station and the computational resource they provide. This information is provided only once at the creation of the station in the hospital.

3.3 Building the global catalog

After creating individual catalogs at each hospital, we build a global catalog encompassing the catalogs from each hospital (recall Figure 1). It results from the union of all the individual catalogs. When the underlying silos are updated, the individual catalogs are re-computed and replaced in the global catalog. Our catalog is intended for end users, e.g., healthcare experts, IT experts. Thanks to our general conceptual model, we are able to display all the data (datasets, features) and their associated profiles in a very simple way. It also eases the creation of visualization charts showing aggregated data. Numeric features can be plotted as histograms where the x-axis is for the values and the y-axis for their frequencies. Categorical features can be shown as pie charts where each slice is a value and its size is proportional to its frequency. Date features can be displayed as bar plots where the axes contain the dates (horizontal) and their frequencies (vertical). Remaining information can be shown as such in the Web interface.

4 Related work

Many platforms have been proposed and developed toward facilitating the cooperation between healthcare centers. For instance, EHDEN [10, 3] and OHDSI [8] are networks of healthcare databases, each mapped to the OMOP common data model [11], a conceptual model for representing observational data. Despite such models being convenient for specific use-cases, they fail at being general and require an (important) effort from experts to map their data to the model. Next, to create federated learning scenarios in a network of interoperable silos, users need catalogs to discover what silos contain (metadata, aggregated data, etc.). EHDEN exposes a Web interface providing a set of pre-defined statistics for each database (number of patients, gender and age distributions, etc.) while OHDSI proposes multiple interfaces, including statistics, ontology exploration and predictions. To generalize catalog modeling, several catalog models have been designed, including DCAT [6] (Data Catalog Vocabulary) and Data Cube Vocabulary [5], both W3C RDF vocabularies. Nevertheless, catalogs have to be (re)developed from scratch for every platform, thus limiting the set of visualizations. On the contrary, we adopt a more general approach based on I-ETL [1] and PADME [12]. Because I-ETL supplies a comprehensive data model not tailored to any specific use-case, we could design a general catalog, able to profile any silo without manual intervention and/or prior knowledge on the data.

5 Conclusion and vision

In this vision paper, we presented our ongoing work toward enabling federated analyses and algorithms across healthcare silos. For this, we have proposed a holistic conceptual model for cataloging silos, as well as a pipeline to create a general catalog, necessary to experts for browsing and designing FL tasks. The main challenge was to design a catalog conceptual model carrying information and statistics both for data and metadata while remaining general. The system is under implementation and will raise many new challenges once finished, including the formulation of queries based on our global models, the query evaluation on aggregated data vs. real data, and the usage of LLMs to ask queries in natural language.

We believe that abstracting current conceptual models dedicated to tailored healthcare use-cases is a promising approach. This allows for their reuse and for the design of use case-agnostic pipelines, which can later be exploited in various settings (without the need to adapt models to the data). It also promotes easy exploration of large and complex silos.

Acknowledgments. This work is supported by the Horizon Europe project BETTER, Grant agreement n. 101136262. We thank all the partners involved in this project for their valuable contributions and feedback.

References

1. Barret, N., et al.: I-ETL: an interoperability-aware health (meta)data pipeline to enable federated analyses. Journal manuscript under revision (2025)
2. Beyan, O., et al.: Distributed analytics on sensitive medical data: the personal health train. Data Intelligence (2020)
3. Blacketer, C., et al.: Using the data quality dashboard to improve the EHDEN network. Applied Sciences (2021)
4. Chen, P.P.S.: The entity-relationship model—toward a unified view of data. ACM transactions on database systems (TODS) (1976)
5. The Data Cube vocabulary. https://www.w3.org/TR/vocab-data-cube/
6. The DCAT vocabulary. https://www.w3.org/TR/vocab-dcat/
7. Gisby, J.S., et al.: Multi-omics identify falling LRRC15 as a COVID-19 severity marker and persistent pro-thrombotic signals in convalescence. Nature Communications (2022)
8. Hripcsak, G., et al.: Observational health data sciences and informatics (OHDSI): opportunities for observational researchers. In: MEDINFO 2015 (2015)
9. Dataset for multi-omics identify LRRC15 as a COVID-19 severity predictor and persistent pro-thrombotic signals in convalescence. https://zenodo.org/records/7410194.
10. Puttmann, D., et al.: Assessing the FAIRness of databases on the EHDEN portal: A case study on two Dutch ICU databases. International Journal of Medical Informatics (2023)
11. Stang, P.E., et al.: Advancing the science for active surveillance: rationale and design for the observational medical outcomes partnership. Annals of internal medicine (2010)
12. Welten, S., et al.: A privacy-preserving distributed analytics platform for health care data. Methods of information in medicine (2022)