Data management for continuous learning in EHR systems

VALERIO BELLANDI, Department of Computer Science, Università degli Studi di Milano, Milano, Italy
PAOLO CERAVOLO, Computer Science, Università degli Studi di Milano, Milano, Italy
JONATAN MAGGESI, Department of Computer Science, Università degli Studi di Milano, Milano, Italy
SAMIRA MAGHOOL, Computer Science, University of Milan, Milano, Italy

To gain a comprehensive understanding of a patient's health, advanced analytics must be applied to the data collected by electronic health record (EHR) systems. However, managing and curating this data requires carefully designed workflows. While digitalization and standardization enable continuous health monitoring, missing data values and technical issues can compromise the consistency and timeliness of the data. In this paper, we propose a workflow for developing prognostic models that leverages the SMART BEAR infrastructure and the capabilities of the Big Data Analytics (BDA) engine to homogenize and harmonize data points. Our workflow improves the quality of the data by evaluating different imputation algorithms and selecting one that maintains the distribution and correlation of features similar to the raw data. We applied this workflow to a subset of the data stored in the SMART BEAR repository and examined its impact on the prediction of emerging health states such as cardiovascular disease and mild depression. We also discussed the possibility of model validation by clinicians in the SMART BEAR project, the transmission of subsequent actions in the decision support system, and the estimation of the required number of data points.

CCS Concepts: • **Information systems → Data management systems**; • **Computing methodologies → Knowledge representation and reasoning**; • **Applied computing → Health informatics**.

Additional Key Words and Phrases: Internet of Things, Electronic Health Records, Data Management, Continuous Learning

1 INTRODUCTION

Electronic Health Record (EHR) systems foster the systematic collection of patients' data in digital format via electronic devices and information systems. The benefits provided by the adoption of EHRs include both organizational and clinical aspects [26]. Clinical decision support systems, computerized order entry systems, and health information exchanges can improve in efficiency with EHRs [15]. Reduced medical errors, improved ability to conduct research, and improved availability of information for patients and clinical staff are societal benefits that EHRs can drive [33].

The acceleration of population aging is a global challenge for healthcare systems [38] and has led lawmakers in the EU, US, and other countries to define recommendations and standards that healthcare providers must follow when implementing EHRs to improve efficiency [19, 40]. This trend is coupled with a growing interest in mobile health (mHealth) monitoring systems. Along with advances in hospital infrastructure, mHealth is setting the stage for the establishment of smart health ecosystems around the world, relying on the availability of data from mobile, wearable, and IoT devices both inside and outside the hospital. These new smart health ecosystems

Authors' Contact Information: Valerio Bellandi, Department of Computer Science, Università degli Studi di Milano, Milano, Lombardia, Italy; e-mail: valerio.bellandi@unimi.it; Paolo Ceravolo, Computer Science, Università degli Studi di Milano, Milano, Lombardia, Italy; e-mail: paolo.ceravolo@unimi.it; Jonatan Maggesi, Department of Computer Science, Università degli Studi di Milano, Milano, Lombardia, Italy; e-mail: jonatan.maggesi@unimi.it; Samira Maghool, Computer Science, University of Milan, Milano, Italy; e-mail: samira.maghool@unimi.it.

ACM 1557-6051/2024/5-ART
https://doi.org/10.1145/3660634

enable the continuous collection of data from everyday life, which will be analyzed to provide the evidence needed to offer personalized interventions [9, 17].

With this convergence, EHR systems are becoming very large in terms of the volume and variety of data they ingest and process, making data management a relevant challenge [6]. Effective data management requires a data strategy and reliable methods to access, integrate, cleanse, store, and prepare data for analysis [25]. The implementation of EHR systems imposes the definition of complete data management procedures with *data quality* and *clinical significance* as key pillars to drive the use of data in enhanced monitoring and diagnostic procedures. This is particularly true for mHealth data, where records may be collected by multiple devices, at different times, and with different levels of quality. Data transmission may be interrupted due to technical or usability issues. Pairing or network issues can limit the availability of IoT devices [27]. Monitoring schedules may be temporarily halted by patients who feel overloaded with their assignments [42]. Different temporal granularity in the collection of observations leads to misaligned time series [14]. Gaps and missing values make time series incomplete, threatening the validity of data analysis [28]. These aspects are well documented in the data quality literature, which emphasizes the importance of implementing tests to verify the *completeness*, *consistency*, and *timeliness* of data and methods to repair or improve data when these dimensions are low [16]. Authors in other research domains are also addressing the data quality and management issue from different perspectives. For instance, *assurance* and *certification techniques* aim to prove that such complex (AI-based [3]) workflows behave as expected and complies with non-functional requirements (e.g., in terms of fairness, accuracy [2, 4], which is out of the scope of the current paper. Furthermore, the relevance of prognostic analytics ultimately depends on the accuracy achieved by the predictive models and the significance of the samples used to train those models [35]. The design of a predictive model involves the evaluation of the complexity of the domain under study, the steadiness of the domain or the size of the samples used to obtain the required accuracy. Ongoing evaluation of models naturally provides the researcher with the means to assess the reproducibility of experimental results, i.e. the extent to which consistent results are obtained when a model is evaluated. Data quality and clinical significance of the data are then two central workflows that an EHR system has to deal with for continuous data acquisition and model adaptation. A priori evaluation of these dimensions is no longer compatible with the goals of today's EHR infrastructures. Software and data management workflows must be designed accordingly.

In this paper, we report the results of the SMART BEAR project in defining a complete data management pipeline for continuous learning in EHR systems. Our solution organizes several data management procedures into automated and modular workflows, which we call the Data Quality Workflow and the Prognostic Model Design Workflow, that allow organizations to foster a culture of continuous improvement. The paper is organized as follows: Section 2 presents the SMART BEAR infrastructure in detail. Section 3 presents a proposed workflow focused on improving data quality and a workflow for designing a predictive model while approaching the sample size required for a valid machine learning (ML) model. In section 4, using the data collected by EHR in the SMART BEAR project, two case scenarios, Cardio Vascular Disease and Mild Depression, are discussed and the results of the implemented workflows are evaluated. We have concluded this paper with concluding remarks and achievements.

2 THE SMART BEAR INFRASTRUCTURE

The SMART BEAR project[1], presents an extensive framework for ongoing, long-term assessments and the monitoring of the health status of elderly individuals. This is achieved through the utilization of wearable devices including smart phones, smart watches, smart thermometers, smart scales, mobile apps, and periodic evaluations conducted by trained personnel and physicians.

[1]https://www.smart-bear.eu/

In complementing Electronic Health Record (EHR) systems, SMART BEAR facilitates continuous monitoring, periodic assessments, data aggregation from various sources, and the provision of both descriptive and predictive analyses.

Efforts to leverage medical/clinical data involve the harmonization of concepts and terms, making the information comprehensible and usable for other clinicians and scientists. A proposed solution involves utilizing the distinct LOINC[2] and SNOMED-CT[3] codes to define observations, encounters, and biological considerations.

The data storage architecture in SMART BEAR adheres to standardized procedures for data acquisition outlined in the "Mapping on Fast Healthcare Interoperability Resources (FHIR)" by [32]. FHIR facilitates the representation and sharing of information among clinicians and organizations in a standardized manner, regardless of the local Electronic Health Record's representation or storage methods, thus advancing interoperability.

The data measured and collected with *SMART BEAR* devices, mobile applications called SB@App that collects temperature and motion via sensors, and questionnaires will be stored in HAPI FHIR repositories using the unified codes. Regarding the integration of questionnaires on the FHIR repository, a generic model is defined in a data model[4]. Moreover, SMART BEAR, takes advantage of HomeHub. The HomeHub is an integral component of the SMART BEAR solution, encompassing both hardware and software aspects. It plays a pivotal role in the deployment of home automation technology within people's residences, serving as the central hub where various devices and sensors can connect and be coordinated.

The main objectives of the SMART BEAR project are continuous and objective monitoring of *quality of life* of elderly people and their ability to live independently [17]. Integrating off-the-shelf smart consumer and medical devices to provide a Smart Health ecosystem, and increasing the efficiency of healthcare delivery while reducing resource waste, are other goals of this project [1]. Considering these objectives, in order to implement efficient and valid analytics in a continuous data acquisition environment, it is required to cure the received data in multiple stages of the data management process [8].

Figure 1 presents the technical infrastructure of the SMART BEAR project [36]. All the data curation procedures are supposed to happen in the Big Data Analytics (BDA) engine that is placed on the SB@Cloud. The SB@Cloud is maintained using the Kubernetes cluster that orchestrates all the services. The cloud is able to run different services including Analysis Workflows (Dashboard and BDA Engine), Data Repository, Decision Support Services, and the Security Component.

The data curation procedures include workflows related to data quality assessment, data preparation, sample size evaluation, and continuous learning.

The BDA engine is tailored to get scalability in terms of the addition of resources to the platform to support the increase in workload. The configuration and deployment of resources are easy and due to the execution on Docker containers, it has high flexibility.

The BDA engine addresses the functionalities required for processing DAWs (Data Analysis Workflows) and storing execution results. It uses a series of suitably configured Open-Source components and a custom-developed one that is responsible for piloting the execution of the various analyses available in the catalog for results storage and presenting them in the dashboard.

In the input data, besides the received data from the SMART BEAR platform, the evaluation of the capability of the infrastructure in integrating other European projects as synergy studies such as Smart4Health[5] and HOLOBALANCE[6] have been a core objective. Smart4Health has been sharing device data and infrastructures involved in low back pain prevention and treatment while the HOLOBALANCE rehabilitation platform is used

[2]https://loinc.org/
[3]http://www.snomed.org/snomed-ct/Use-SNOMED-CT
[4]https://www.hl7.org/fhir/questionnaireresponse.html
[5]https://smart4health.eu
[6]https://holobalance.eu

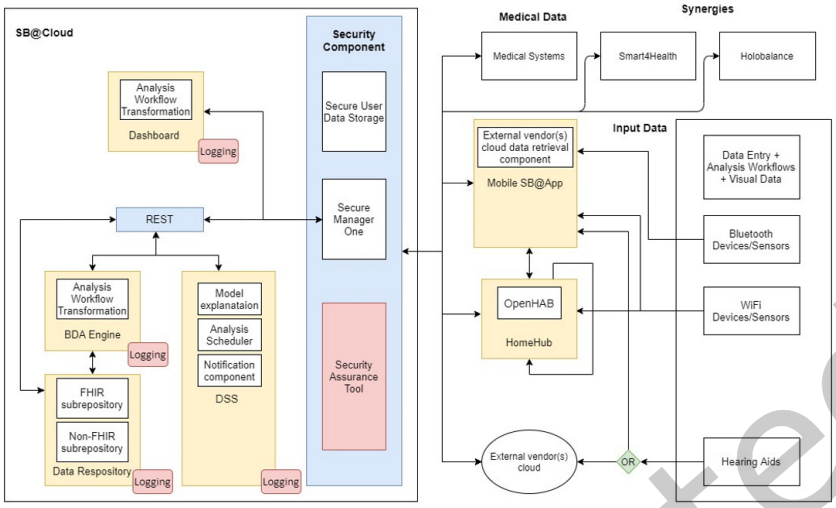

Fig. 1. An overview of the SMART BEAR infrastructure as presented in [36]. It contains different components such as Big Data Analysis Engine, Security Component, Decision Support System, Dashboard, and Data Repository. The received data from home sensors and synergies are depicted on the right side of the picture.

to support participants with balance disorders by projecting a virtual holographic physiotherapist through the Holobox to guide the patient to the daily exercise regime to carry out.

2.1 BDA engine components

The components used by the platform are as follows.

- **Apache Hadoop**[7]**:** From the Hadoop ecosystem, an exclusively distributed file system (HDFS) is utilized for storing the extracted data from the FHIR repository and transforming it in a tabular format appropriately.
- **Apache Hive Metastore**[8]**:** It contains all the information regarding the databases, the tables, and the relationships between them. This is especially useful for handling data that is transformed and saved on the distributed file system. The Metastore allows interoperability between the various components that need to access the data and enables to launch SQL-like queries using Trino and Spark.
- **Apache Spark**[9]**:** This is the component that allows data management, data engineering, and ML tasks on a large dataset. Thanks to its abilities, the data are accessible through the Metastore considering the HDFS data similar to tables in a SQL database, simplifying life for those who have to develop the analytics and guaranteeing the necessary scalability for the platform.
- **Trino**[10]**:** It is a distributed SQL query engine designed to query large data sets distributed over one or more heterogeneous data sources. It is designed to handle data warehousing and analytics: data analysis, aggregating large amounts of data, and producing reports (OLAP). It could be leveraged for ETL transformations without the overhead of Spark. Furthermore, Trino with the proper configuration can run SQL queries directly on tables stored in HDFS.

[7]https://hadoop.apache.org
[8]https://hive.apache.org
[9]https://spark.apache.org
[10]https://trino.io

- **Apache Airflow**[11]**:** It can develop, schedule, and monitor batch-oriented workflows. Airflow's extensible Python framework enables us to build workflows using different technologies, combined with docker, it can run custom images with all the tools needed to run analytic tasks.
- **BDA API:** It is internally developed for providing a REST-type interface to the dashboard and other platform components in order to be able to interact with workflows and save/retrieve the results of previous executions. The BDA API includes a catalog of atomic analytics that could be composed in workflows the engine schedule automatically or in dependence on specific events.
- **Delta Lake**[12]**:** It is an open-source storage framework that enables building a Lakehouse architecture [5]. It is located on top of Hadoop providing ACID Transactions, and scalable metadata handling while unifying streaming and batch data processing on top of existing data lakes like HDFS .
- **Apache Zeppelin**[13]**:** It is a Web-based tool that enables users to create interactive data analysis, prototype some of the analytics that should be translated into a proper workflow, and share preliminary results in a collaborative environment for the data scientist.

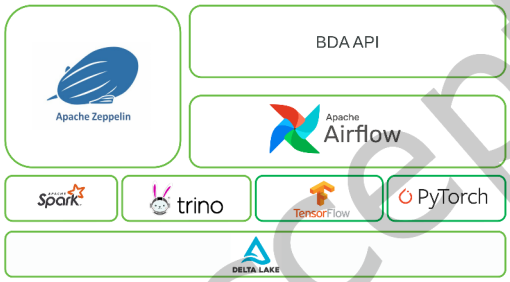

Fig. 2. The schematic view on the Big Data engine used for the analysis tasks of the SMART BEAR project. Components are depicted hierarchically considering the layered steps to proceed from data extraction to BDA API.

2.2 Data storing and data management implemented by BDA

Exported data from various storage sources, such as the FHIR repository, is transformed through a series of steps and stored in the format requested by Delta Lake. This format allows for building a Lakehouse architecture. The Lakehouse is an open architecture that combines the best elements of data lakes and data warehouses. As some of the key features of this kind of architecture, we can name: (i) ACID transactions, (ii) schema enforcement, (iii) business intelligence support, (iv) openness, (v) support for diverse workloads, and (vi) support for structured and unstructured data [11].

Through Delta Lake, we support all these requirements and offer a real version management process for data in our environment. With the *Time Travel* component we can specify the version of a table. This allows us to be able to relaunch an analysis on a specific version of the data in order to be able to make a comparison between the results of different snapshots. It also provides a mechanism to replicate a result, and audit the data and the obtained results.

[11]https://airflow.apache.org
[12]https://delta.io
[13]https://zeppelin.apache.org

2.3 Data analysis tasks

Data analysis tasks are mainly performed using Spark and Python libraries that can operate on data in Delta Lake format. Depending on the complexity of the workflows, these tasks could be composed of several steps. In some cases, it is easier to query and prepare data through Spark and then use other more specific libraries or tools to perform the required analytics. With Apache Airflow we can create DAGs (Directed Acyclic Graphs) composed of different tasks and use HDFS as a distributed file system to save the data needed for the various steps. Furthermore, with this flexibility, if necessary, it is possible to use also ready Docker images containing all the necessary tools for creating models and performing analytics. The main requirement therefore always remains to be able to read data from a distributed file system such as HDFS.

2.4 Data flow management and workflow orchestration

The most complex job in data flow management is querying the FHIR repository to export the data from the repository and insert it into the tables in Delta Lake. The workflow, that runs each day, is therefore composed of different tasks which can be roughly divided as follows: (i) export from FHIR, (ii) flattening of the data, and (iii) upsert in Delta Table. To export the data, the possibilities offered by the FHIR REST API are exploited, using the Bulk Data Export API. Therefore it is also possible to request to export data that has been inserted/modified in the repository from the last time the export process was successful. The data exported in *ndjson* format are then saved on HDFS, the next step is to bring them in a flat type format, i.e., as if they were data belonging to a common SQL table. All of this is orchestrated using Apache Airflow as the main engine. Using Airflow, it is possible to configure the workflows to be executed according to predefined intervals, either with some already predefined, or in very specific cases it is possible to execute them also using schedules defined through an interval defined through a Cron-type expression.

2.5 Data visualization

For the data visualization part, the BDA Engine mainly makes use of two different tools: Apache Echarts[14] to display the results of the analytics in the Dashboard interface and Apache Zeppelin to allow data scientists to carry out data exploration and try to create examples of analytics which will be deployed in production. Apache Echarts is an open-sourced JavaScript visualization tool that can run on Web Browser and mobile devices, it also provides an excellent library of basic charts and the possibility to extend or customize according to the needs of our output. There are many available components, among the most used there are certainly: (i) *Datazoom* that is used for zooming a specific area, which enables users to investigate data in detail, get an overview of the data, or get rid of outlier points. (ii) *Timeline* which provides functions like switching and playing between multiple Echarts. (iii) *Toolbox* that contains some functionalities like the export to PNG and zooming. (iv) *Legend* that shows symbol, color, and name of different series. The user can click legends to toggle displaying series in the chart.

2.6 Platform management

Since all the tools used in the BDA Engine have been containerized, to manage the entire platform the tool we rely on is the deployment platform itself, i.e., Kubernetes (K8s). *K8s* is the system that we use to handle scaling, automatic system deployment and manage all the containerized applications that we use in the Cloud [29]. All the components described above have their basic configurations saved as ConfigMap inside the Smart Bear repository dedicated to the BDA Engine while passwords and sensitive data are saved as Secrets. For some components, it was decided to use Helm Charts [15] to deploy as the Spark cluster, while for others ad-hoc deployments are

[14]https://echarts.apache.org
[15]https://helm.sh

used. Then the final configurations should be done using the WEB UI provided by the tool. These administration UIs are not available to all users, but only to system administrators who can also modify the deployment of the various tools and manage the resources needed to keep the infrastructure fast enough.

2.7 Analytics Orchestration

The analytics that can be performed with custom configuration by clinicians through the dashboard is exposed and managed through the BDA REST API. These APIs expose a catalog of the available analytics with their various configuration parameters and their default values. Each time an analytic is launched these parameters are passed to Airflow which takes care of managing the execution. In the UI, the data scientists can therefore create a new notebook or clinicians can use the already implemented notebooks to launch different analyses (or perhaps even the same one, but with different parameters) and then analyze the results which are returned at the end of the execution. In case of problems, the system administrator can always check the various runs using the Airflow Web UI and check in the logs where the problem occurred, if it is due to a wrong configuration, lack of resources, or something else. Obviously, for the clinician or data scientist, the management of the platform is totally transparent, in fact, the configuration of the capacity of the scheduler and the number of workers available are tasks left to the system administrator who must find the right compromise between the available resources and the speed of execution requested by the end user.

3 IMPLEMENTED WORKFLOWS

This section illustrates two major workflows supported by the SMART BEAR BDA engine. Each of them is composed of sub-tasks and sub-workflows we discuss. Their implementation is realized to support *continuous learning* and *validation* of prognostic models. SMART BEAR complements an EHR system by providing continuous monitoring, data gathering, and analysis. Newly arrived data must be ingested, verified, and possibly improved. Their temporal and domain validity must be checked. Data representation must be homogenized and normalized to have a common representation format. A virtualized version of data must be made available in sandbox environments allowing researchers to explore them without affecting the access to the storage. Data imputation procedures are necessary to improve the completeness and standardization of the stored data sets. The representativeness and significance of a data set in training a prognostic model must be verified before exploiting its recommendations to notify the users of the systems. The models stored in the system must be continuously adapted to newly acquired data and their performance monitored.

3.1 Data Quality Workflow

In general terms, data quality consists of multiple-step implementation taking care of different aspects to consider for improving the accessibility, usability, and efficiency of analytic results [16]. We require our BDA to explore different traits of data quality. In particular, considering the distribution of missing values, we will define a continuous method for learning this distribution and select, accordingly, the best imputation algorithm. The execution of this workflow is a precondition for training accurate and reliable prognostic models.

- Step 1: *Data Homogenization.*
  Clinical data are usually collected from a range of sources, and each data origin point has its unique structure and format which requires much effort in unifying the concepts and terms to make the data understandable and usable by both clinicians and scientists. As previously mentioned in Section 2.2, SMART BEAR creates the interoperable data model for structural homogenization while for contextual homogenization purposes takes advantage of the identified LOINC[16] and SNOMED-CT[17] codes in defining observations, encounters,

[16] https://loinc.org
[17] http://www.snomed.org/snomed-ct/Use-SNOMED-CT

and biological considerations that will be stored in FHIR repositories [12].
In the context of the SMART BEAR project, a data model for mapping the questionnaires according to the FHIR principles has been developed. According to this model, an FHIR questionnaire template requires resources where the URL, name, and title shall have a value while the version might have a value. This way, the questionnaires are mapped on the FHIR data model.

- Step 2: *Data Organizing and Restructuring.*
  The common input format of data analytics and machine learning procedures is a d-dimensional vector where each dimension is a measurable piece of data, a.k.a feature or attribute. As discussed in Section 2.4, in the FHIR repository, semi-structured data are stored in JSON format and contain both arrays and nested objects. In order to bring them into a tabular format compatible with the vectors ingested by data analytic algorithms, an ad-hoc script is created using Apache Spark consists of the following steps:
  – Flatten all the properties, which means that each property and object nested in the JSON creates a column by concatenating the name of the properties with those that are children of the main property.
  – Create a new row for each element of an array if the element contains arrays within the structure. In this case, a new row is created in the resulting table for each element of this array.
- Step 3: *Data Transformation.*
  To move data into a common data set, users require three operations: Extraction, Transformation, and Loading (ETL). By extraction, we query and gather the required data in the original data set; while by Transformation we could apply multiple changes such as Data Filtering, Data Mapping, Data Deduplication, Data Cleansing, Derived Variables, Data Sorting, or Ordering. All these operations are needed to proceed incrementally as the SMART BEAR is running and new data is arriving. For Extraction and Loading purposes, the BDA engine takes advantage of the Trino and Delta lake as we mentioned in the BDA engine components Section 2.1
- Step 4: *Data Virtualization.*
  Data virtualization creates a layer where users can access, retrieve, and manipulate data as needed. It brings all the information together in one virtual location, allowing real-time access with no need to perform ETL. Data virtualization is often more cost-effective and accurate. SMART BEAR uses the Apache Zeppelin environment for this purpose. Zeppelin is a web-based notebook that enables data-driven, interactive data analytics and collaborative documents. Using Zeppelin data scientists can implement step 3 and proceed with the rest of the procedure serving the homogenization and organizing while exploring the available data. Notebooks created in this way are a prototype to obtain feedback and then move on to the analytics deployment phase in the system. After the notebook is finalized by the experts we can start to use the code and test it, the accepted code can be released and included in the BDA catalog to make it accessible via the BDA API. Through Zeppelin, it is possible to launch Python, Spark, SQL Query, or R scripts. This is possible by the different interpreters that are provided by the system and that are launched using Docker images containing the required components and libraries already configured to execute the snippets.
- Step 5: *Data Normalization.*
  Data normalization is the organizing of data to make them more similar in form and scale across all records and fields. It increases the cohesion of entry types leading to the cleansing the unstructured data, redundancies, and duplicates. Data normalization is basically the mapping functions unifying the scales of numerical values without loss of generality. For example, normalization by max/min values and z-score are advised regarding the scope of analytics.
- Step 6: *Data imputation.*
  The sparse multivariate temporal series collected by smart devices in smart-health projects requires specific methodologies to leverage feature sparsity by effectively studying the correlation between features, the achieved data quality levels, and appropriate techniques for filling the missing values to be adopted. The

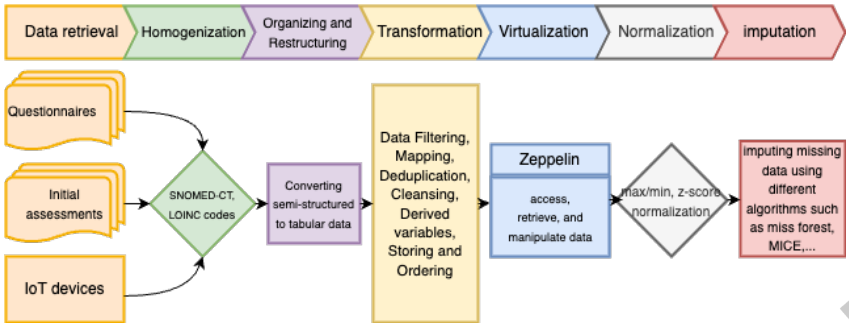

Fig. 3. Schamtic view on data quality workflow

proportion of missing data only provides limited information about the bias and efficiency gains that can be made from data imputation. The recent literature clarified that data imputation needs to be anticipated by correlation analysis to verify the pattern followed by missing values [20, 23, 24, 30, 37]. This implies that researchers should consider whether all the variables related to missingness can plausibly be included in the imputation model to limit bias and improve accuracy.
An additional challenge of imputation techniques in longitudinal studies is related to the consistent integration of temporal series and scalar features in imputation models [7]. Figure 3 illustrates the steps of this workflow schematically.

3.2 Workflow for designing a prognostic model

The healthcare delivery system relies on complex decision-making, and various modeling tools are proposed to assist in ensuring optimal patient care. Prognostic models, among these tools, aim to improve predictions based on retrospective data, aiding clinicians in decision-making.

However, designing a prognostic model is a challenging issue, particularly in cases where data are insufficient or incomplete. In such cases, continuous learning is crucial to ensure the accuracy of models over time, as newly arrived data can improve the assigned weights of existing features and adjust data imputation procedures. This ongoing learning process also helps to validate the consistency of a trained model as time passes. To address these issues, a workflow has been developed that fulfills the requirements for designing a prognostic model. This workflow involves multiple steps that can be executed separately, with a validation step proposed at the end to ensure the accuracy of the model. The steps followed for designing a diagnostic model are illustrated in Figure 4 and presented below.

Step 1: *Feature extraction.* The extraction of suitable features is crucial in managing the complexity of a problem when dealing with heterogeneous data types in machine learning (ML) models. This involves unifying a dataset containing diverse data types like time series, categorical data (e.g., gender), and numerical data (e.g., age). The goal is to create a unified form and temporal dimension for these inputs in a d-dimensional feature space required for ML model training. Various algorithms are available, including those extracting static statistical features and others based on embedding algorithms that maintain the temporal relevance of data points while abstracting records.[7].

Step 2: *Sample size approximation* . Predicting the volume of needed data points for validating the ML model is one of the crucial aspects of initiating a study with limited resources. The SMART BEAR project proposes a three-stage procedure for defining, implementing, and validating prognostic models. The first stage involves

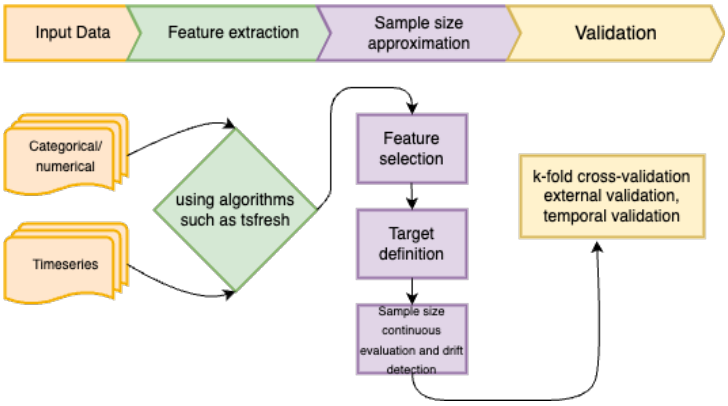

Fig. 4. Schematic view on workflow for designing a diagnostic model

homogenizing multidimensional data into a single-valued representation. The second stage employs correlation analysis to identify patterns in missing data, aiding integration into ML models for imputation and predictions. The third stage studies the fraction of missing information to predict its impact on prognostic models.

The sample size must be evaluated with a temporal plan to define a credible timing for getting realistic volumes of data. We cannot plan using complex models if the data size is not appropriate. For example, experimental evidence from the literature [41] shows Random Forest is reliable with a sample size of 200 times the number of features in the feature space. This means once we know the sample size, we can also identify the dimension of the feature space to be used and the modeling tool to select.

- Feature selection (the variables to be used in the learning process): for a specific model, we need to select specific features out of the existing measurements that are carried out in the SMART BEAR project.
- Target definition (the variables to be predicted): as we will discuss more specifically the target of the model should be defined, in other words, which is the output of the model.
- Sample size continuous evaluation and drift detection: in a running project, we repeatedly run the sample size approximation since new data brings new insight into the feature space.

Step 3: *Validation*. Regarding the ML models, the new dataset could play the test set role while the validation to find the appropriate parameters leverages the k-fold cross-validation algorithm. As an external validation, temporal validation is also could be proposed. Besides the validation, in projects such as SMART BEAR, accuracy is not the only assessed parameter as it may be biased and some approval from the clinicians is a requirement. The rationale behind this decision is the active exchanges with the clinicians, for the adoption of a specific model or sample size not only the accuracy of the model should be stable on all cross-validated folds (low-variation) but the clinicians should approve the suitability of the output and the extracted information' relevance.

4 PRACTICAL IMPLEMENTATION OF WORKFLOWS IN THE SMART BEAR EXPERIMENTAL RESULTS

The SMART BEAR project assesses and evaluates the proposed workflows using the initial dataset derived from assessments, observations, questionnaires, and devices in the Pilot of the Pilots (PoP). The Pilot of the Pilots (PoP) is a smaller subset of one of the pilots recruiting participants for the SMART BEAR project in which more than 100 patients have been recruited and monitored to demonstrate the project concept feasibility before the kick-off of the large-scale ones. It had the main objective to test the first release of the SMART BEAR infrastructure as a

prerequisite for the additional five pilots, and to show the synergies implemented with other European projects as recommended by the European Commission, highlighting cooperation and interaction between complementary solutions that are evolving at different paces.

By focusing on elements unique to continuous data collection scenarios, our discussion now shifts to the critical areas of data imputation and sample size evaluation. Delving into these aspects allows us to address the nuances and challenges inherent in continuous data collection scenarios. By exploring the intricacies of data imputation, we aim to improve our understanding of how to effectively impute missing or incomplete data to ensure the robustness and reliability of our datasets. At the same time, sample size assessment becomes paramount, providing insight into the adequacy of our data collection efforts for meaningful statistical analyses.

4.1 Data Imputation in SMART BEAR

In the SMART BEAR project, like other workflows, imputation is a recurrent process as new data sets are added to repositories. This iterative implementation enhances accuracy by updating values based on dynamic measurements. Considering the dataset's diverse values (numerical, categorical, ordinal), homogenization steps are followed. Numerical values undergo z-score normalization using mean and standard deviation. Categorical values are mapped to numerical values, treating null values as an extra category through a mapping function. For instance, with a one-to-one mapping function, 'yes/no/null' becomes '0/1/2'. Following imputation, reverse mapping and one-hot encoding are applied to the dataset for training machine learning models.

Various algorithms for data imputation have been proposed in the literature. Considering the type of data, the distribution and proportion of missing values, and the sensitivity of the data, we need to choose an algorithm that returns values closest to the real ones. Considering the inaccessibility of the ground truth data, the evaluation of the selected algorithm could be a big challenge for a data scientist.

4.1.1 Proposed approaches for data imputation. In this workflow, considering the size of the received data set and the correlation matrix of existing variables, we propose two generic approaches in order to select the proper algorithm for data imputation:

- **Heuristic:** While the literature has clarified it is not easy to classify the distribution of missing data and it requires having a deep understanding of the domain but most of the methods start from the assumption that distribution is Missing At Random (MAR) and a correlation with other observed data is possible. In this approach, different "Multivariate" algorithms will be performed on the data set imputing the missing values, such as regression, stochastic regression, XGboost, MICE, and miss forest are considered for the imputation purpose.
- **Experimental:** A priori classification of the distribution of missing data is not only difficult but also impossible once there are multiple sources of missingness. Therefore, testing experimentally the different methods seems a necessity. In the experimental approach, as is depicted schematically in Figure 5, we proposethe following. 1) Compute the correlation matrix in the data set. 2) Apply multiple methods of data imputation considering the correlated variables. 3) Verify which one is producing the correlation matrix most similar to the original correlation matrix or in other words the absolute difference of two correlation matrices before and after imputation is converging to zero. 4) Using it apply again multiple methods. 5) Iterate until the variation of the correlation matrix before and after imputation is negligible. The pseudo-code that has generated this approach is described in detail in Algorithm 1.

Algorithm 1 Experimental continuous imputation workflow

Require: N-column data frame ($\{x_i\}$), Threshold (T_0), List of Univariate algorithms (U), List of Multivariate algorithms (M)
Ensure: Calculate correlation matrix (c_{ij})
  for $i <= N$ **do**
    if $c_{ij} < T_0$ **then**
      apply Univariate methods
      for $k \in U$ **do**
        Calculate the sum of absolute differences of correlation row for the ith variables before and after imputation (abs diff(k_i))
      end for
      Choose the U_k with least abs diff(k_i)
    else
      apply Multivariate methods
      for $l \in M$ **do**
        Calculate the sum of absolute differences of correlation row for the ith variables before and after imputation (abs diff(l_i))
        Choose the M_k with least abs diff(l_i)
      end for
    end if
  end for
  Fully imputed data frame
  for $i <= N$ **do**
    Return x_i to its original form, keep the rest imputed
    Calculate correlation matrix (c'_{ij})
    if $c'_{ij} < T_0$ **then**
      apply Univariate methods
      for $k' \in U$ **do**
        Calculate the sum of absolute differences of correlation row for the ith variables before and after imputation (abs diff(k'_i))
      end for
      Choose the U_k with least abs diff(k'_i)
    else
      apply Multivariate methods
      for $l' \in M$ **do**
        Calculate the sum of the absolute difference of correlation row for the ith variables before and after imputation (abs diff(l'_i))
      end for
      Choose the M_k with least abs diff(l'_i)
    end if
  end for

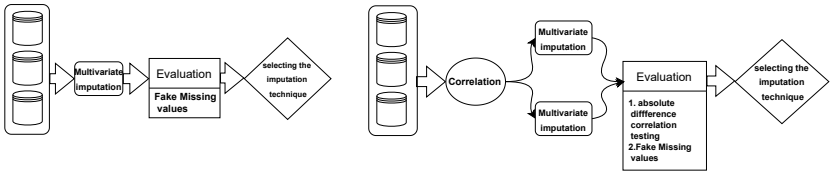

Fig. 5. A schematic view on the heuristic (left) and experimental (right) data imputation workflow.

4.1.2 Algorithm selection based on the absolute difference of correlation matrix: In this method, the fact that the data distribution should not be changed by the imputation process is verified by comparing the data correlation before and after the imputation. Figure 6 shows an example of changes in the correlation matrix due to imputation over a set of features from the SMART BEAR repository. According to this example, even though systolic and diastolic blood pressure are highly correlated, the applied imputation algorithm does not change this correlation, so this could be a trustworthy solution.

Fig. 6. Algorithm selection based on the absolute difference of correlation matrix

4.1.3 Algorithm selection based on the minimum error evaluating fake missing values. After applying several imputation algorithms, in order to decide which one has the best accuracy on each dataset, we need to evaluate how accurately they predict the missing values. Since the ground truth data are missing, for evaluation purposes we manually try to exclude some existing real data points called *fake missing values* and predict them by comparing them with the real values and calculating the root mean square error (RMSE), mean absolute error (MAE) and absolute error (AE). Figure 7 shows the calculated error types in terms of different proportions of fake missing values.

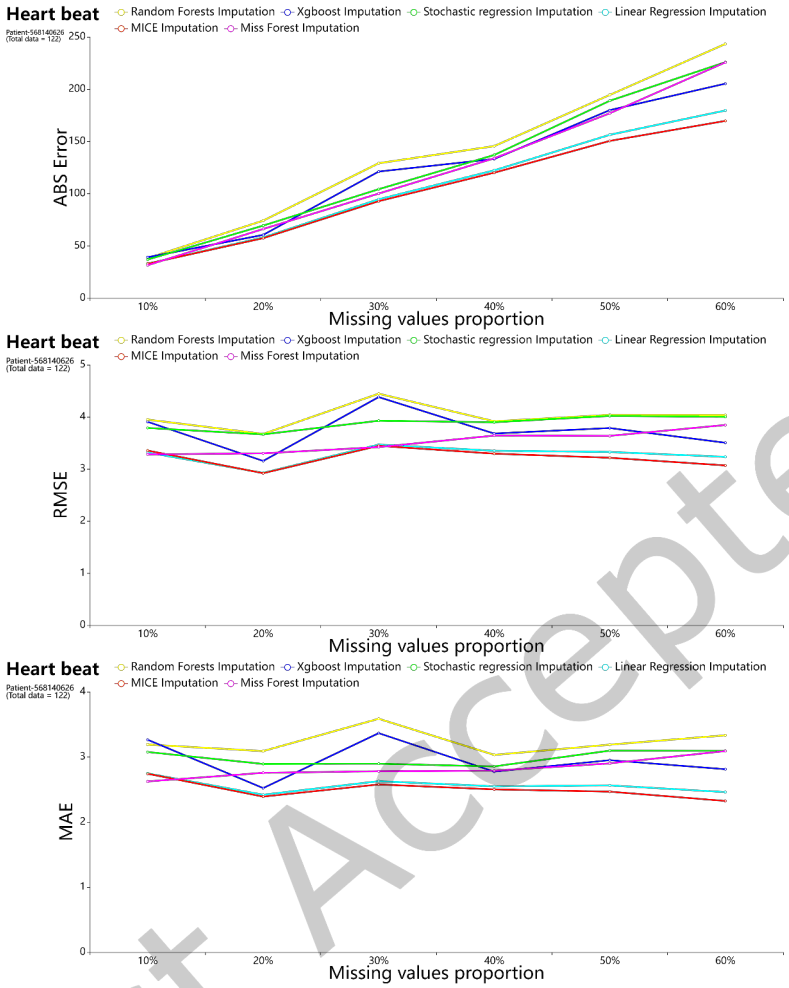

Fig. 7. Using the HearBeat data from SMART BEAR data repository, we have imputed the *missing values* using state-of-the-art algorithms. Considering the different proportions of fake missing values (on the x-axis), we evaluate the *absolute error*, *Root mean square error* and, *Mean average error* between the predicted data and the ground truth data.

4.1.4 Continuous Learning. Implementing continuous learning, or in other words, ensuring that workflows run frequently during the SMART BEAR study, is critical to the success of the pilots. With the arrival of new data sets, there is an opportunity to use them to modify trained models and fine-tune relevant hyperparameters. As previously discussed in the context of the imputation workflow, it is essential to evaluate imputed values while maintaining the assumption that the distribution of the fully imputed dataset is the same as the initial one. In line with this assumption, we have automated this workflow to select the best algorithm based on minimizing the deviation of the feature correlation. As a result, the analytics offered by the SMART BEAR dashboard can now delineate the trajectories that patients follow as they improve or deteriorate in specific capacities. They are also critical in identifying risk thresholds. This insight is invaluable for making informed decisions and improving patient care outcomes.

4.2 Designing a prognostic model

In this section, we illustrate the steps we take to design predictive models. Sample size approximation can be performed as an automatic step that runs periodically as new data is collected.

4.2.1 Feature extraction. Due to different data types, such as numerical and time series, we need to unify the input data to run an ML model. To do this, we map the time series to a set of static numerical features. Using the *Time series feature extraction* algorithms such as *tsfresh*[18], a list of features of the given time series is returned, sorted by the importance weight of them in forming the trend. For the current study, due to the lack of data points, we limit the number of time series features to avoid over-fitting in model training. Table 1 represents a short list of features proposed by *tsfresh*, sorted by the importance value assigned by the algorithm for the systolic blood pressure time series with a time interval of 12 hours (twice a day). Combining these features with the main variables, we can proceed with the modeling of ML algorithms. As we will discuss later in the implementation section, adding time series features can increase the accuracy of the models. An important aspect we need to consider is the imputation of missing values in the time series. Since we have already discussed the data imputation process in section 4.1, we also need to find the best imputation algorithms for time series. For this purpose, assuming that the distribution of values should remain unaffected by imputation, we investigate whether the weight of the extracted features changes significantly before and after imputation. In Table 1, the third and fourth columns show the value of the corresponding features, while the last column shows the absolute error (AE) caused by the imputation process.

List of time series features	feature	value before imputation	value after imputation	AE
1	Systolic_blood|_pressure_variance _larger_than_standard_deviation	1	1	0
2	Systolic_blood_pressure_has _duplicate_max	0	0	0
3	Systolic_blood_pressure_has _duplicate_min	0	0	0
4	Systolic_blood_pressure_has _duplicate	1	1	0
5	Systolic_blood_pressure_sum _values	1148	15740.8	1452.8

Table 1. Sample of extracted features by *tsfresh* and the comparison of values before and after data imputation.

4.2.2 Sample size approximation. Similarly to the data imputation approaches, we take two approaches for sample size approximation:

- **Heuristic:** Due to the fact that the ML algorithms need a large amount of data to train reasonably accurate models, and according to the existing literature, at least 200 times the number of features is needed as training data set for Random Forest classifiers [41]. Considering simple models with few features, it would be possible to train an ML model, although there is a trade-off between model accuracy and overfitting with very few (or very large) data.
- **Experimental:** In studies such as SMART BEAR, which contains many features, it is not feasible to acquire such a large dataset with limited resources to recruit many participants. In this sense, we propose an experimental approach to approximate the sample size needed to train ML models. In this approach, as schematically shown in Figure 8, we consider the following steps. 1) We start by imputing the dataset using the best imputation algorithms evaluated and mentioned above . 2) We augment the dataset to reach a secondary, extended data set. 3) We execute an Automated Machine Learning (AutoML) procedure. AutoML tools are designed to automate the time-consuming and complex aspects of machine learning, allowing

[18]https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html

users to build and deploy models more efficiently. These tools leverage algorithms and heuristics to search through the model space, selecting the best-performing models for a given task. By reducing the need for extensive manual intervention, AutoML enables individuals without deep machine learning expertise to harness the power of machine learning for their specific applications. 4) For validation purposes, we take advantage of internal validation, such as cross-validation, and external like temporal validation using the dataset of the SMART BEAR project with different time intervals 5) Once clinicians approve the accuracy of the model, we stop augmentation and extract the most important features. Going back to the inspiration of the heuristic approach, multiplying the number of features by 200 and roughly calculating the number of data points needed for the accurate model. 6) In case that clinicians do not approve the accuracy of the model we continue with the augmentation to reach the aimed accuracy.

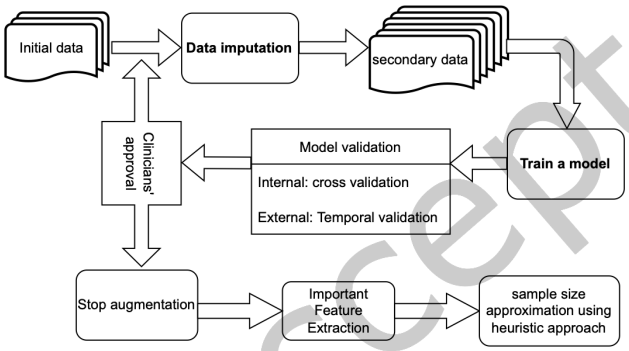

Fig. 8. A schematic view on the sample size approximation workflow.

4.3 Use case scenarios

To design a prognostic model that uses the available features in the dataset to predict a possible outcome and to evaluate the role of features in the final state, we propose to study the usability of our workflows on practical use case scenarios. In this way, the effectiveness of implemented steps on individual features will be well represented in the holistic evaluation score. In the literature, some studies prove the correlations between features and final states of patients [34]. Instead of testing a single hypothesis, once we have a handful of features to consider, we use ML algorithms to train a model and predict the final state.

4.3.1 Cardio Vascular Disease case study. In the SMART BEAR project, the understudied cohort includes participants with cardiovascular disease (CVD). To enhance analysis efficiency, clinicians provide a list of key measurements related to CVD development, enabling the training of a classifier for predicting CVD. This example evaluates our imputation workflow's classifier accuracy. Despite the limited data points at this early stage of collection, the results, while not definitive, suggest the potential utility of our proposed data imputation workflow for Electronic Health Record (EHR) analysis. Following the steps in Section 3.1, we obtain a normalized and fully imputed dataset using an algorithm with minimal error in the presence of simulated missing values. Table 2 presents scoring metrics from training an XGBOOST algorithm with the H2O package AutoML models [22].

	Before imputation		After imputation	
Metric	**threshold**	**value**	**threshold**	**value**
max f1	0.79406	0.90909	0.4124311	0.960000
max f2 (weighted harmonic mean of precision and recall)	0.45997	0.95238	0.4124311	0.9836066
max accuracy	0.79406	0.86666	0.5211698	0.933333
max precision	0.86615	1.0	0.9482347	1.0
max recall	0.45997	1.0	0.4124311	1.0
max specificity (True Negative Rate)	0.86615	1.0	0.9482347	1.0
max absolute_mcc	0.79406	0.70710	0.5211698	0.8291562
max min_per_class_accuracy	0.79406	0.83333	0.5211698	0.9166667
max mean_per_class_accuracy	0.79406	0.91666	0.5211698	0.9583333
max True Negative Rate	0.86615	1.0	0.9482347	1.0
max False Negative Rate	0.86615	0.91666	0.9482347	0.9166667
max False Positive Rate	0.45997	1.0	0.2188251	1.0
max True Positive Rate	0.45997	1.0	0.4124311	1.0

Table 2. Evaluation metrics resulted from training XGBOOST classifier using AutoML on the normalized data set **before** and **after** imputation. Considering the chosen threshold, the values of metrics are reported in column *value* on the test set.

Employing H2O ensures hyperparameter tuning consistency. The ROC curve [18] in Figure 11 illustrates model performance before and after the imputation process, showcasing the impact on predictive accuracy.

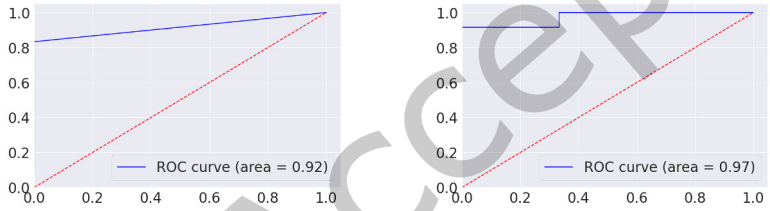

Fig. 9. Comparison of ROC curves resulted from the same data set, using H2O, before (a), and after (b) imputation.

4.3.2 Mild Depression case study. Since there are very few solutions on the market dedicated to early detection and/or prevention in the field of mental health, we actively focus on the depression factors that could be considered as early signs or risk factors of clinical depression, before the older person enters full-blown depression. Similar to the CVD case scenario, we are implementing our model in close exchange with clinicians based on their suggested features. Regarding the Mild Depression case study, we try to evaluate the importance of sample size by following the workflow proposed in section 3.2. In this workflow, starting from the already fully computed dataset resulting from the data quality workflow in section 3.1, we add 10% to the data. The following table shows a comparison of evaluation metrics during the data augmentation process to approximate the appropriate sample size for the specified model accuracy. As an intermediate validation check, if the acquired accuracy is acceptable to the clinicians, we stop the augmentation at this step, otherwise we continue the procedure by adding another 10% (Table 3) to the dataset. After reaching agreement with the clinicians on the level of accuracy, we extract the most important features of the model. Likely in our scenario, they are 10, then looking at the heuristic approach, we need 200 times the number of features for a valid model. The summary of the process is reported in Table 3, where the evaluation matrices of the trained model on the data set before augmentation, augmented by 10%, 20% and 40% are presented. During the augmentation process, we not only observed an increase in the average *accuracy*, but also in most of the validation folds, the improvements are clear. Figure 10 shows the ROC curves resulting from the classifier. An insightful step after training the classifier is feature extraction which not only

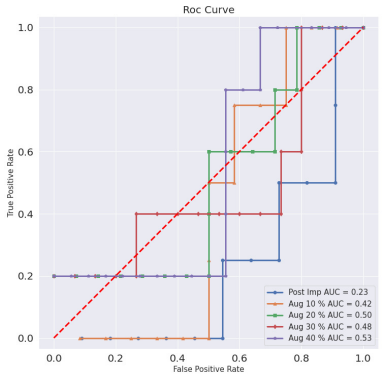

Fig. 10. Comparison of the improvements in the prediction problem using Gradient Boost Machine considering the different proportions of augmentation of the original dataset.

helps in explainability but also in following the reproducibility of models during the augmentation process in our workflow and during data collection in the real case scenarios. Figure 11, contains four stages of augmentation using the SHAP values. With these sub-figures, we witness the most important features are quasi-fixed during the augmentation process which could be proof of the choice of the augmentation method that does not change the weight of the features in the original dataset. The highly feature values are colored in red while moving toward blue color, features losing their significance. Regarding the SHAP values, the more the values the more the feature has added a value in the output of the model.

Evaluation matrices on 5-fold cross-validated before augmentation						
metrics	**mean**	**cv_1**	**cv_2**	**cv_3**	**cv_4**	**cv_5**
accuracy	0.78823 ± 0.1288	0.82352	0.94117	0.82352	0.76470	0.58823
f1	0.54272 ± 0.2111	0.4	0.8	0.4	0.75	0.36363
precision	0.4111 ± 0.2053	0.25	0.66666	0.33333	0.6	0.22222
recall	0.9 ± 0.2236	1.0	1.0	0.5	1.0	1.0
Evaluation matrices on 5-fold cross validated after augmentation by 10%						
accuracy	0.83976 ± 0.135	0.89473	0.94736	0.94736	0.63157	0.77777
f1	0.67142 ± 0.126	0.66666	0.85714	0.66666	0.66666	0.5
precision	0.67435 ± 0.3071	0.5	1.0	1.0	0.53846	0.33333
recall	0.825 ± 0.2091	1.0	0.75	0.5	0.875	1.0
Evaluation matrices on 5-fold cross validated after augmentation by 20%						
accuracy	0.89047 ± 0.1087	0.95238	0.95	0.9	0.7	0.95
f1	0.77619 ± 0.0887	0.85714	0.85714	0.66666	0.7	0.8
precision	0.86060 ± 0.1911	1.0	1.0	0.66666	0.63636	1.0
recall	0.72222] ± 0.05196	0.75	0.75	0.66666	0.77777	0.66666
Evaluation matrices on 5-fold cross-validated after augmentation by 40%						
accuracy	0.91449 ± 0.05328	0.91666	0.91666	0.95652	0.82608	0.95652
f1	0.85396 ± 0.02286	0.83333	0.85714	0.85714	0.83333	0.88888
precision	0.84670 ± 0.14132	0.71428	0.75	1.0	0.76923	1.0
recall	0.89181 ± 0.114289	1.0	0.8	1.0	0.90909	0.8

Table 3. The output of a Gradient Boost Machine, used as a classifier from the H2O algorithms. The mean and standard deviation of matrices and each fold from a 5-fold cross-validated data set are presented. From the top, the results are related to before augmentation, and after the augmentation process respectively the initial data set is augmented by 10%, 20%, and 40%.

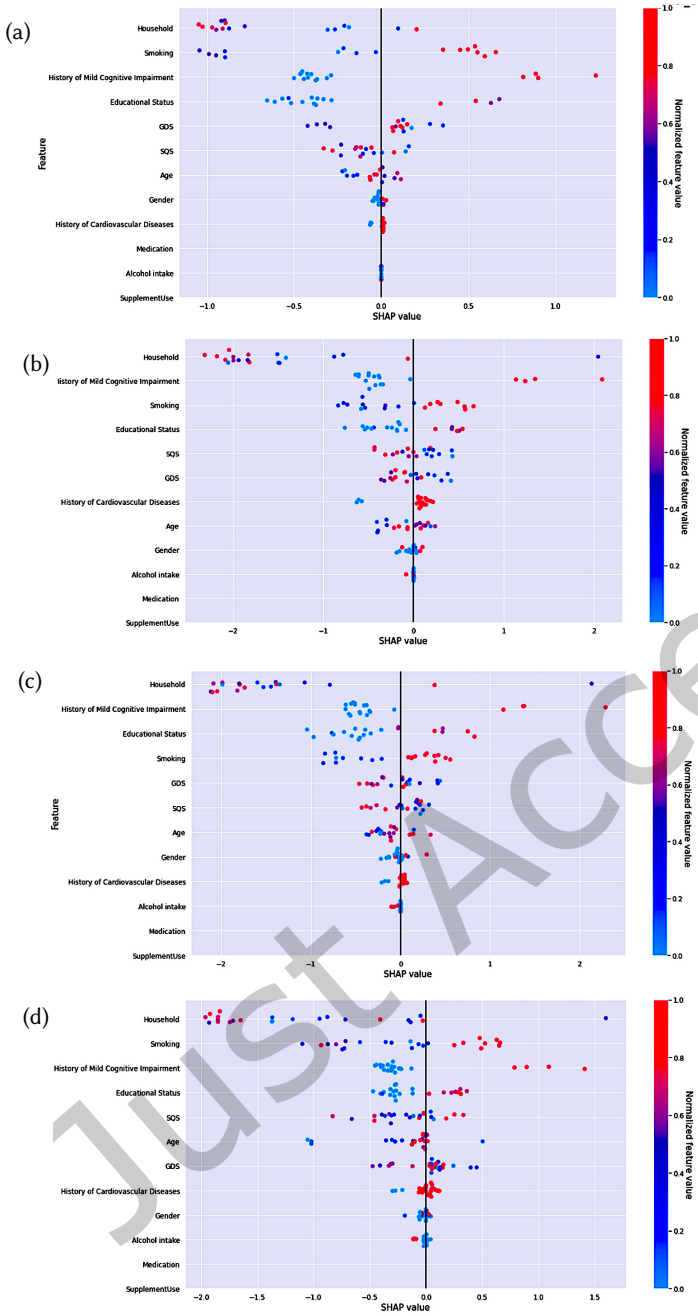

Fig. 11. Important features of the GBM classifier to predict the state of Mild Depression in the patients using proposed features by clinicians. On the top left, the chart refers to the cases before augmentation, and on the top right to the case with 10% augmentation. On the second row-left 20% augmentation while on right the case with 40% augmentation features is presented.

5 RELATED WORKS

Nowadays, due to the drastic increase in the volume of data, the incorporation of modeling tools for decision-making in the healthcare domain is becoming inevitable. ML models for clinical studies mostly rely on supervised learning. In supervised learning, the model is developed using gold standards defined by a clinical expert, for example, a chart review containing 1000 patients with and without Cardio Vascular Diseases (CVD), aiming to identify the existing pattern between patients with CVD vs those without it. In this process, while some hypotheses are claimed by clinicians the researcher tries to prove those hypotheses using ML algorithms but rejecting them needs more clinical and theoretical validation and evidence [10]. On the other hand, unsupervised ML models can also be used when there is a need to phenotype multiple conditions. These models are not generally as accurate as supervised models but enable high-throughput variables over a handful of thousands of them with improved accuracy. Moreover, unsupervised models have been used to build clinical models to predict disease courses, optimize diagnostics, and target treatment. Moreover, unsupervised pattern recognition analyses identify subgroups of patient-patient similarity in a high-dimensional or graph-based space. Whilst EHR systems are constantly producing and recording data, leveraging Machine Learning algorithms for creating prognostic and predictive models has implications for patients, caregivers, and healthcare facilities for cost management purposes. In this direction, one crucial point to take into account is the quality of collected data from EHR systems that several studies focused on it. Detecting the issues systematically is an interesting problem that is explored in [39]. The authors have found patterns in the condition domain and investigated the processes that shape them suggesting data quality issues influenced by system-wide factors that affect individual concept frequencies. The most general patterns identified in the literature are Missing Not At Random (MNAR), Missing At Random (MAR), and Missing Completely At Random (MCAR) [21]. MNAR, means there is a relationship between the propensity of a value to be missing and its values. For example, people with the lowest education are not answering questionnaires including the questionnaire on their educational courses. MAR, means missing values are not related to other missing values but are related to observed values. For example, men are more likely to report their weight than women. MCAR, means there is no relationship between missing values and any other values. Nothing makes some data more likely to be missing than others. For example, blood pressure records are missing randomly, due to user ignorance or of charge battery. Recent studies found that, compared to restricting the analysis to individuals with complete data, imputation techniques improved the accuracy of predictions at any proportion of missing data [23]. This implies that researchers should consider whether all the variables related to missingness can plausibly be included in the imputation model to limit bias and improve accuracy. While in the longitudinal studies topics covered include reliability, validity, sampling, aggregation, and the correspondence between theory and method; more specific, practical issues in longitudinal research, such as the drop-out problem and issues of confidentiality are also addressed [13, 31], the automation of this procedure is still missing. Moreover, due to the sensitivity of the health care domain, not only a deep knowledge of the process is needed but also continuous evaluation of the curation strategy should be considered.

6 CONCLUSIONS

The paper focuses on the potential challenges of continous learning in today's EHR. To address this issue, we leveraged the capabilities of the SMART BEAR BDA engine to develop two workflows: the *Data Quality Workflow* and the *Workflow for Designing a Prognostic Model*. These workflows enable continuous and automated data curation, as well as sample size approximation, to ensure that valid machine learning (ML) tasks can be implemented in the BDA engine. To evaluate the effectiveness of our proposed workflows, we ran prediction problems in two case scenarios using data from the SMART BEAR Pilot of Pilots (PoP).

Our main goal was to improve the quality of the dataset, making it more applicable to valid models, while also investigating the clinical significance of the models by providing explainable results. The diagram in Figure

12 presents the applicability of our methodology as a workflow consisting of different passages of the dataset collected from patients to provide useful insights to them.

Our results suggest that using these workflows not only increases the accuracy of ML tasks, but also allows us to reduce the feature space of a problem, thus avoiding overfitting and improving model performance. By extracting the available dataset from the SMART BEAR PoP, we were able to demonstrate the effectiveness of our workflows in achieving higher quality datasets for use in valid models.

Considering the case scenario of cardiovascular disease (section 4.3.1), as an approach to test the effectiveness of the proposed data quality workflow before and after its implementation, we proposed a classification model to predict cardiovascular disease status. For this model, we queried the relevant measurements suggested by clinicians in the SMART BEAR project. Using the XGBoot model, we observed (in Table 2) the performance improvements in most of the evaluation metrics. Moreover, considering the case scenario of Mild Depression (section 4.3.2), in Figure 7, the ROC curve has been improved after the implementation of this workflow. It should also be noted that although we observe improvements in the performance of the predictive models since the dataset is still small, we have studied the performance after increasing the dataset and the results still prove our statement.

In order to communicate the required number of participants for reliable results in the resource-constrained SMART BEAR project, we increase the robustness of the results by extending the data set to avoid potential overfitting. This ensures validation of results by accepting results when samples are sufficient and rejecting them when the model lacks generalizability to the extended dataset. This experimental approach contrasts with heuristic methods that rely on the number of classes or features in the feature space to determine correctness.

Overall, this work represents a significant step forward in automating the data curation process and sample size approximation, which are critical components of effective *continuous machine learning*. By continuously improving the quality of the dataset and continuously validating the generated models, we can ensure more accurate, reliable, and clinically meaningful results. Given the ongoing nature of the SMART BEAR project, we're expanding workflow automation as new datasets come in. Future plans include additional validation methods, including temporal validation, using automated workflows that ensure model reproducibility. As we explore different scenarios, such as frailty, we envision implementing other predictive models in future efforts.

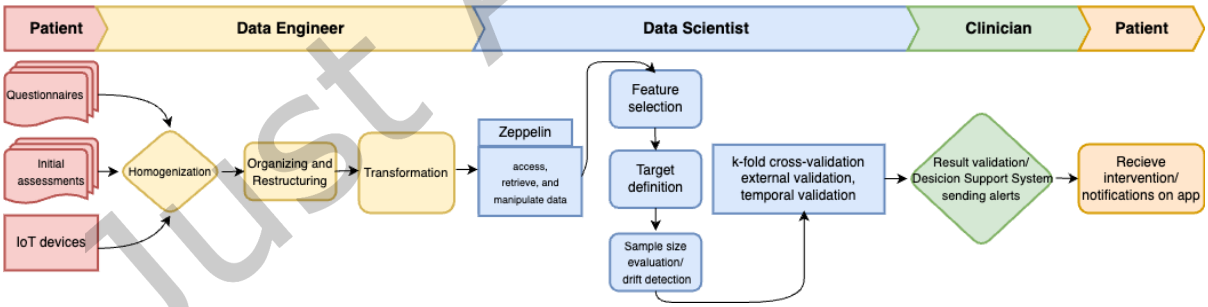

Fig. 12. Data Flow: From patient to patient; a schematic view of the flow of data generated by patients, developed by technical partners, and analyzed by data scientists. The output is validated by clinicians and sent to the decision support system to notify patients when interventions are needed.

ACKNOWLEDGMENTS

This work is partially supported by i) the Università degli Studi di Milano within the program "Piano di sostegno alla ricerca", ii) the MUSA – Multilayered Urban Sustainability Action – project, funded by the European Union –

NextGenerationEU, under the National Recovery and Resilience Plan (NRRP) Mission 4 Component 2 Investment Line 1.5: Strengthening of research structures and creation of R&D "innovation ecosystems", set up of "territorial leaders in R&D, iii) the project SERICS (PE00000014) under the MUR NRRP funded by the EU - NextGenerationEU and, iv) the European Union's Horizon 2020 research and innovation program under the SMART BEAR project, grant agreement No 857172.

REFERENCES

[1] C Agostinho, A Pimenta, M Marques, KM Tsiouris, F Kalatzis, C Nikitas, E Iliadou, M Occhipinti, I Kouris, D Koutsouris, et al. 2022. Healthier and Independent Living of the Elderly: Interoperability in a Cross-Project Pilot. In *CEUR Workshop Proceedings*. CEUR, 1–4.

[2] Marco Anisetti, Claudio A. Ardagna, and Nicola Bena. 2023. Multi-Dimensional Certification of Modern Distributed Systems. *IEEE TSC* 16, 3 (2023).

[3] Marco Anisetti, Claudio A. Ardagna, Nicola Bena, and Ernesto Damiani. 2023. Rethinking Certification for Trustworthy Machine-Learning-Based Applications. *IEEE IC* 27, 6 (2023).

[4] Claudio A. Ardagna and Nicola Bena. 2023. Non-Functional Certification of Modern Distributed Systems: A Research Manifesto. In *Proc. of IEEE SSE 2023*. Chicago, IL, USA.

[5] Michael Armbrust, Ali Ghodsi, Reynold Xin, and Matei Zaharia. 2021. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In *Proceedings of CIDR*, Vol. 8.

[6] Antonia Azzini, Sylvio Barbon Jr, Valerio Bellandi, Tiziana Catarci, Paolo Ceravolo, Philippe Cudré-Mauroux, Samira Maghool, Jaroslav Pokorny, Monica Scannapieco, Florence Sedes, et al. 2021. Advances in data management in the big data era. In *Advancing Research in Information and Communication Technology: IFIP's Exciting First 60+ Years, Views from the Technical Committees and Working Groups*. Springer, 99–126.

[7] Francesco Bagattini, Isak Karlsson, Jonathan Rebane, and Panagiotis Papapetrou. 2019. A classification framework for exploiting sparse multi-variate temporal features with application to adverse drug event detection in medical records. *BMC medical informatics and decision making* 19, 1 (2019), 1–20.

[8] Valerio Bellandi. 2023. A Big Data Infrastructure in Support of Healthy and Independent Living: A Real Case Application. *Intelligent Systems Reference Library* 229 (2023), 95 – 134.

[9] Valerio Bellandi, Paolo Ceravolo, Ernesto Damiani, Samira Maghool, Ioannis Basdekis, Matteo Cesari, Eleftheria Iliadou, and Mircea Dan Marzan. 2022. A methodology to engineering continuous monitoring of intrinsic capacity for elderly people. *Complex & Intelligent Systems* (2022), 3953–3971. https://doi.org/10.1007/s40747-022-00775-w

[10] Valerio Bellandi, Paolo Ceravolo, Ernesto Damiani, and Stefano Siccardi. 2022. Smart Healthcare, IoT and Machine Learning: A Complete Survey. *Intelligent Systems Reference Library* 212 (2022), 307 – 330. https://doi.org/10.1007/978-3-030-83620-7_13

[11] Valerio Bellandi, Paolo Ceravolo, Samira Maghool, and Stefano Siccardi. 2022. Toward a general framework for multimodal big data analysis. *Big Data* 10, 5 (2022), 408–424.

[12] Duane Bender and Kamran Sartipi. 2013. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In *Proceedings of the 26th IEEE international symposium on computer-based medical systems*. IEEE, 326–331.

[13] LR Bergman. 1996. Measurement and data quality in longitudinal research. *European Child & Adolescent Psychiatry* 5 (1996), 28–32.

[14] Munish Bhatia and Sandeep K Sood. 2019. Exploring temporal analytics in fog-cloud architecture for smart office healthcare. *Mobile Networks and Applications* 24, 4 (2019), 1392–1410.

[15] David Blumenthal and Marilyn Tavenner. 2010. The "meaningful use" regulation for electronic health records. *New England Journal of Medicine* 363, 6 (2010), 501–504.

[16] Paolo Ceravolo and Emanuele Bellini. 2019. Towards configurable composite data quality assessment. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, Vol. 1. IEEE, 249–257.

[17] Alessia Cristiano, Sara De Silvestri, Stela Musteata, Alberto Sanna, Diana Trojaniello, Valerio Bellandi, Paolo Ceravolo, and Matteo Cesari. 2021. IoT Platform for Ageing Society: the SMART BEAR Project. In *The Thirteenth International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2021)*. IARIA.

[18] Bradley J Erickson and Felipe Kitamura. 2021. Magician's corner: 9. Performance metrics for machine learning models. , e200126 pages.

[19] European Commission. 2022. Exchange of electronic health records across the EU. https://digital-strategy.ec.europa.eu/en/policies/electronic-health-records. Accessed: 2022-12-04.

[20] Chenguang Fang and Chen Wang. 2020. Time series data imputation: A survey on deep learning approaches. *arXiv preprint arXiv:2011.11347* (2020).

[21] Venkat Gudivada, Amy Apon, and Junhua Ding. 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software* 10, 1 (2017), 1–20.

[22] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* 212 (2021), 106622.

[23] Rachael A Hughes, Jon Heron, Jonathan AC Sterne, and Kate Tilling. 2019. Accounting for missing data in statistical analyses: multiple imputation is not always the answer. *International journal of epidemiology* 48, 4 (2019), 1294–1304.
[24] Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. 2017. When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. *BMC medical research methodology* 17, 1 (2017), 1–10.
[25] Clemens Scott Kruse, Anna Stein, Heather Thomas, and Harmander Kaur. 2018. The use of electronic health records to support population health: a systematic review of the literature. *Journal of medical systems* 42, 11 (2018), 1–16.
[26] Daniel Lewkowicz, Attila Wohlbrandt, and Erwin Boettinger. 2020. Economic impact of clinical decision support interventions based on electronic health records. *BMC health services research* 20, 1 (2020), 1–12.
[27] Xiaopeng Li, Qiang Zeng, Lannan Luo, and Tongbo Luo. 2020. T2pair: Secure and usable pairing for heterogeneous iot devices. In *Proceedings of the 2020 acm sigsac conference on computer and communications security*. 309–323.
[28] Yuehua Liu, Tharam Dillon, Wenjin Yu, Wenny Rahayu, and Fahed Mostafa. 2020. Missing value imputation for industrial IoT sensor data with large gaps. *IEEE Internet of Things Journal* 7, 8 (2020), 6855–6867.
[29] Marko Luksa. 2017. *Kubernetes in action*. Simon and Schuster.
[30] Paul Madley-Dowd, Rachael Hughes, Kate Tilling, and Jon Heron. 2019. The proportion of missing data should not be used to guide decisions on multiple imputation. *Journal of clinical epidemiology* 110 (2019), 63–73.
[31] David Magnusson and Lars R Bergman. 1990. *Data quality in longitudinal research*. Vol. 3. Cambridge University Press.
[32] Joshua C Mandel, David A Kreda, Kenneth D Mandl, Isaac S Kohane, and Rachel B Ramoni. 2016. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association* 23, 5 (2016), 899–908.
[33] Nir Menachemi and Taleah H Collum. 2011. Benefits and drawbacks of electronic health record systems. *Risk management and healthcare policy* 4 (2011), 47.
[34] Isaac Moshe, Yannik Terhorst, Kennedy Opoku Asare, Lasse Bosse Sander, Denzil Ferreira, Harald Baumeister, David C Mohr, and Laura Pulkki-Råback. 2021. Predicting symptoms of depression and anxiety using smartphone and wearable data. *Frontiers in psychiatry* 12 (2021), 625247.
[35] Chetta Ngamjarus. 2016. n4Studies: Sample size calculation for an epidemiological study on a smart device. *Siriraj Medical Journal* 68, 3 (2016), 160–170.
[36] Vadim Peretokin, Ioannis Basdekis, Ioannis Kouris, Jonatan Maggesi, Mario Sicuranza, Qiqi Su, Alberto Acebes, Anca Bucur, Vinod Jaswanth Roy Mukkala, Konstantin Pozdniakov, et al. 2022. Overview of the SMART-BEAR Technical Infrastructure. In *Proceedings of the 8th International Conference on Information and Communication Technologies for Ageing Well and e-Health-ICT4AWE,*. SciTePress, 117–125.
[37] PS Raja and KJSC Thangavel. 2020. Missing value imputation using unsupervised machine learning techniques. *Soft Computing* 24, 6 (2020), 4361–4392.
[38] Ewa Rudnicka, Paulina Napierała, Agnieszka Podfigurna, Błażej Męczekalski, Roman Smolarczyk, and Monika Grymowicz. 2020. The World Health Organization (WHO) approach to healthy ageing. *Maturitas* 139 (2020), 6–11.
[39] Casey N Ta and Chunhua Weng. 2019. Detecting systemic data quality issues in electronic health records. *Studies in health technology and informatics* 264 (2019), 383.
[40] U.S. Centers for Medicare & Medicaid Services. 2021. Electronic Health Records. https://www.cms.gov/Medicare/E-Health/EHealthRecords. Accessed: 2021-01-12.
[41] Tjeerd van der Ploeg, Peter C Austin, and Ewout W Steyerberg. 2014. Modern modelling techniques are data hungry: a simulation study for predicting dichotomous endpoints. *BMC medical research methodology* 14, 1 (2014), 1–13.
[42] Upkar Varshney. 2013. Smart medication management system and multiple interventions for medication adherence. *Decision Support Systems* 55, 2 (2013), 538–551.

Received 8 April 2023; revised 19 March 2024; accepted 23 March 2024