Computer Science and Information Systems 20(4):1591–1638 https://doi.org/10.2298/CSIS230331062B

DG_Summ: A Schema-Driven Approach for personalized Summarizing Heterogeneous Data Graphs

Amal Beldi[1,2], Salma Sassi[2], Richard Chbeir[2] and Abderrazek Jemai[1,3]

[1] Tunis El Manar University, Faculty of Mathematical Physical and Natural Sciences of Tunis, SERCOM Laboratory, 1068 Tunis, Tunisia
amal.beldi@univ-pau.fr
[2] University Pau & Pays Adour, LIUPPA, Anglet, 64600, France
salma.tissaoui@univ-pau.fr
richard.chbeir@univ-pau.fr
[3] Carthage University, Polytechnic School of Tunisia, SERCOM Laboratory, INSAT, 1080, Tunis, Tunisia
abderrazekjemai@yahoo.co.uk

Abstract. Advances in computing resources have enabled the processing of vast amounts of data. However, identifying trends in such data remains challenging for humans, especially in fields like medicine and social networks. These challenges make it difficult to process, analyze, and visualize the data. In this context, graph summarization has emerged as an effective framework aiming to facilitate the identification of structure and meaning in data. The problem of graph summarization has been studied in the literature and many approaches for static contexts are proposed to summarize the graph. These approaches provide a compressed version of the graph that removes many details while retaining its essential structure. However, they are computationally prohibitive and do not scale to large graphs in terms of both structure and content. Additionally, there is no framework providing summarization of mixed sources with the goal of creating a dynamic, syntactic, and semantic data summary. In this paper, our key contribution is focused on modeling data graphs, summarizing data from multiple sources using a schema-driven approach, and visualizing the graph summary version according to the needs of each user. We demonstrate this approach through a case study on the use of the E-health domain.

Keywords: Heterogenous data, labeled graph, Graph summarization,operation, structure, content, versioning

1. Introduction

Graph datasets, such as those found in social networks, astronomy, and bioinformatics, are a common type of big data application. They consist of large−scale interconnected nodes and edges, which provide a more natural representation of the data. By querying and analyzing the relationships between these entities, it is possible to uncover valuable insights into a wide range of phenomena. This type of analysis can lead to profound discoveries and deep understandings of complex systems. However, due to sheer volume, complexity, and temporal characteristics, building a concise representation (i.e., summary) helps to understand these datasets as well as to formulate queries in a meaningful way. Graph

summarization has emerged as a popular research topic to data in recent years. Its goal is to simplify the task of identifying the structure and significance of data represented in a graph format. A summary of a graph is a brief representation of the original graph that can be utilized for a variety of purposes. For example, it can reduce the number of bits required to encode the original graph, or enable more complex database-style operations to summarize graphs with scalable resolution that can be adjusted interactively[69]. Graph summarization is a useful technique that offers several advantages and has a wide range of applications, including interactive and exploratory analysis[24], approximate query processing [25], visualization [28], data-driven Visual graph query interface construction [35] and distributed Graph Systems among others. However, current approaches for summarizing graphs in static contexts, such as modularity-based community detection[32], spectral clustering[62], graph−cut algorithms [21] exist to summarize the graph in terms of its communities, but lack explicit ordering and only provide groupings without characterizing the subgraphs or offering a clear understanding of the outputs[6]. While these approaches are effective in summarizing the structure of graphs, they do not offer comprehensive characterization of the outputs. The increasing prevalence of dynamic graphs and streams has created a need to analyze their evolving properties over time, leading to a renewed interest in developing graph synopsis construction methods that can accurately summarize their characteristics [67]. However, existing approaches often lack explicit ordering in groupings, leaving users with limited time and no clear starting point for understanding their data. Moreover, these approaches are typically designed for static contexts and lack direct dynamic counterparts. Although some algorithms, such as the one presented in [62], can operate in dynamic settings, they only focus on identifying static patterns that persist over multiple time steps. Thus, there is currently no framework available that can provide a comprehensive summary of mixed-source information while also creating a syntactic and semantic summary of the data. This paper addresses the challenges associated with generating a comprehensive summary that captures both the structure and content of mixed-source data, as well as the relationships and interactions with past data. In particular, the paper focuses on addressing the following challenges:

- **Challenge 1**: How can we generate a summary that integrates data from multiple sources in various formats, such as text, video, and images?
- **Challenge 2**: What methods can be developed to generate user-oriented, semantic-based summaries that are tailored to specific information needs and retrieval challenges?
- **Challenge 3**: How can we ensure that a summary can effectively analyze and capture the changing nature of real data over time, while still providing a concise and informative representation of the underlying data?

The remainder of this paper is structured as follows. In Section 2, a scenario in the healthcare domain is presented along with the limitations and requirements for effective summarization. Section 3 provides an overview of models for data representation and summarization techniques. Section 4 describes and discusses related works on graph summarization approaches, with a particular focus on electronic health record summarization approaches to validate the limitations of our proposed scenario. Section 5 outlines the architecture of our approach, followed by a formalism of the Data Graph in Section 6 and a detailed description of the summary process in Section 7. Section 8 provides explana-

tioncof the experiments and Section 9 describes evaluates our approach both qualitatively and quantitatively. Section 10 conclude with perspectives for future work.

2. Motivating scenario

The healthcare scenario involves a pregnant woman who may develop gestational diabetes, which is more likely if she is over 25, overweight, has a family history of diabetes, had gestational diabetes in a past pregnancy, is prediabetic, has high blood pressure, or had COVID-19. To control his gestational diabetes and understand the patient's history, she needs to communicate regularly with her General Practitioner (GP) and share data such as glucose level, temperature, blood pressure, and location. A Type 2 Diabetes Monitoring System called T2DM system shown in Figure 1, is installed to control and monitor the patient's gestational diabetes and is based on patient data including patient history, Electronic Health Records (EHR), clinical documents, and data from Medical Devices (MD).

It enables the GP to summarize patient data and receive automatic alarms that provide feedback on the patient's health status. The system is capable of monitoring gestational diabetes-related parameters using MDs. However these MDs are heterogeneous in terms of deployment, computing capabilities, and communication protocols. This generates a massive volume of heterogeneous and ambiguous data, which makes it challenging to analyze patient data efficiently and understand the patient's medical history, current situation, treatment status, and dynamic contact network.

To better analyze and monitor the patient's gestational diabetes, there is a need for improvements in integrating these systems for better interoperability and data standardization. Additionally, better data management and analysis tools are required to handle the large volume of heterogeneous and ambiguous data generated by MDs. To enhance patient care, there is a requirement for an integrated approach that amalgamates data from various sources, including electronic health records, clinical documents, and medical devices. However, the current systems are inadequate, and there is still room for improvement as they are unable to meet certain requirements. We imagine that a GP requests the system to answer certain queries, and we present below some queries that T2DM, as well as existing solutions, remain unable to handle:

- **Query 1**: Providing a concise and comprehensive summary of the patient's medical history, with a focus on their Type 2 Diabetes, the GP would need to utilize existing medical systems and technologies to present the patient's demographic data, medical and surgical histories, family history, and current medication list simultaneously in a synthesized manner [61], [1], [48]. However, currently, it is not feasible to condense the most recent blood work results (or older results if there have been significant changes) or biological test results into a summarized format.
- **Query 2**: Generating a graphical representation that summarizes the changes in blood sugar levels over the past two months. In order to measure blood sugar levels, the T2DM uses some existing works [26], which presents the results in either a text or a table format. The numbers within the brackets refer to specific blood sugar level measurements, which are taken at different times of the day. However, currently, there is no way to create a graphical or textual summary of the changes in blood sugar levels

over the past two months. This means that it is not possible to visualize how blood sugar levels have fluctuated during this period.

- **Query 3**: Displaying a real-time summary of data from all MDs. This means that the system would need to collect and process data continuously as it is generated by each MD. However, at present, there is no way to generate a real-time synthesis of data, which means that it is not possible to display a constantly updated overview of the data collected from multiple MDs.
- **Query 4**: Displaying a periodic summary of blood sugar and temperature evolution, with an indication of the visited area for each day. This would involve collecting data on blood sugar and temperature levels for each day and presenting it in a graphical format, with an added feature of indicating the geographical location of each data point. For example, a map could be used to show where the data was collected from. However, currently, there is no system in place to create this type of periodic synthesis of data, so it would need to be developed.

Table 1. Queries and Requirements for Handling Clinician Cases in our proposed scenario

Query	Summarization Objective	Data Source	Summary format	Limitations	Requirments
Query 1	Demographic data, medical and surgical histories, family history, and current medication list	EHR data and Biological tests, Blood test	Multiple summarization	Unable to summarize recent blood work results or biological test results	Aggregating the blood work or biological test results over a certain period of time
Query 2	Changes in blood sugar levels over the past two months	Blood sugar level measurements	Graphical or textual summary	Unable to generate graphical or textual summary	Applying arithmetic operators and generating Graphical or tabular summary
Query 3	Real time synthetis of data	Medical Devices	Constantly Updated Graphic	Unable to generate a real time synthesis of data	Summarizing data in real-time Interpreting and displaying it in graphical or tabular form
Query 4	Evolution of Blood sugar and temperature within geographic location information	Daily blood sugar and temperature measurements	Graphical representation with geographic markers	Unable to create periodic synthesis of data	Creating a periodic summary as well as the location and the time of each data points.

The existing Type 2 Diabetes Monitoring System (T2DM) is capable of monitoring various parameters using Medical Devices (MDs), but the heterogeneous nature of these

MDs generates a large volume of ambiguous and heterogeneous data, making it challenging to analyze and understand patient data efficiently. To enhance patient care, there is a need for an integrated approach that combines data from various sources, including electronic health records, clinical documents, and medical devices. However, current systems are inadequate, and there is still room for improvement in terms of data standardization, interoperability, and data management and analysis tools. In this scenario, as shown in table1, several queries remain unanswered by T2DM and existing solutions. For instance, there is a need for a concise and comprehensive summary of the patient's medical history, including recent blood work and biological test results in a synthesized format. Additionally, graphical representations that summarize the changes in blood sugar levels over time and display a periodic summary of blood sugar and temperature evolution, with an indication of the visited area for each day, are not currently available. To address these needs, the system should be able to aggregate data over a certain period of time, apply arithmetic operators to numerical values to generate charts or summary tables depicting the evolution of certain numerical measures. It should be able to collect and process data in real−time, combine and synthesize it to generate a real-time summary, and interpret data to present it in the form of charts, tables, or other visualizations. Also, the system should be capable of creating a graphical output that displays the location of each data point, which can help in understanding the patient's health status and treatment progress. In summary, this healthcare scenario highlights the need for an integrated approach to patient care that combines data from various sources and utilizes better data management and analysis tools. The development of such a system would enhance patient care and facilitate efficient communication between healthcare providers and patients, leading to better treatment outcomes. Thus, towards building such approach, the following challenges emerge:

- **Challenge 1**: How can we generate a summary that integrates data from multiple sources in various formats, such as text, video, and images?
- **Challenge 2**: What methods can be developed to generate user-oriented, semantic-based summaries that are tailored to specific information needs and retrieval challenges?
- **Challenge 3**: How can we ensure that a summary can effectively analyze and capture the changing nature of real data over time, while still providing a concise and informative representation of the underlying data?

3. Background

In this section, we initially provide a concise explanation of various models used for data representation to justify the utilization of graph−based models in the paper. Afterward, we offer a comprehensive overview of diverse techniques for graph summarization.

3.1. Models for data representation

There are various models for representing data including network [17]relational databases [29],RDF and ontology[16] and graph models [7]. Each of these models has its own strengths and weaknesses, and the choice of model depends on the specific requirements and characteristics of the data being analyzed. Network models [17] are commonly used

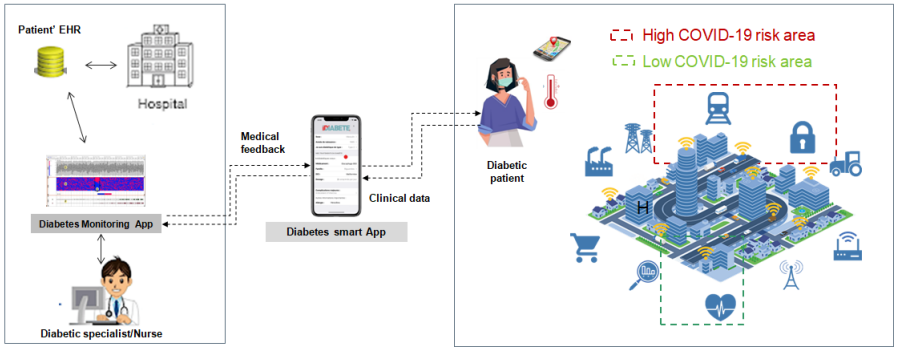

Fig. 1. Type 2 Diabetes Monitoring system (T2DM system)

to represent complex relationships between entities, such as social networks, biological networks, and transportation networks. These models are useful for analyzing the structure and connectivity of the network, as well as identifying patterns and communities within the network.Relational databases [29] are used to represent data in a structured format, with well-defined relationships between tables and fields. These models are useful for querying and analyzing large datasets, and for enforcing data consistency and integrity. RDF and ontology models [16] are used to represent knowledge and metadata in a structured format, with a focus on semantics and relationships between concepts. These models are useful for representing complex and heterogeneous data, and for enabling interoperability and data sharing across different domains. Graph models [7], on the other hand, are a general-purpose model for representing data as a set of nodes and edges. This model is particularly useful for representing and analyzing data that is highly heterogeneous and contains complex relationships between entities. Graph models can handle both structured and unstructured data, and can be easily extended to incorporate additional information or attributes. In the context of data summarization, graph models are often the preferred choice for representing and analyzing large and complex datasets that contain both structured and unstructured data. This is because graph models can capture complex relationships between entities and provide a flexible and scalable framework for data analysis. Additionally, graph models can be used to represent data from different sources and domains, making them well-suited for analyzing data from heterogeneous databases. Therefore, graph models are a suitable choice for summarizing structured and unstructured data from highly heterogeneous databases. In this paper, we have chosen to use the graph-based model for data summarization because it can effectively represent and analyze large and complex datasets that contain both structured and unstructured data. The graph model's ability to capture complex relationships between entities and provide a scalable framework for data analysis also makes it a suitable choice for analyzing data from heterogeneous databases. Moreover, using a graph database can offer significant advantages when dealing with connected data. Graph databases have superior query performance compared to relational databases and other NoSQL alternatives, which is essential for efficiently processing large amounts of data during summarization tasks. Graph databases also offer greater flexibility for adapting to changing needs, as they allow for the addition of relationships, node types, and properties without modify-

ing existing queries. Additionally, their schema-less nature can reduce ambiguity in the domain model and enable more accurate and precise modeling during summarization. Finally, using a graph database can speed up the design-to-delivery process, as developers can create a data model on a whiteboard without having to worry about translating it to a set of tables, ultimately leading to faster delivery of summarization results.

3.2. Summarization Methods

We provide here a background of knowledge about the existing methods related to Data summarization. Many domains have extensively studied summarization, including text analysis, network traffic monitoring, the financial domain, the health sector, and many others. The summarization problem arises in a variety of data analysis tasks and application domains. A variety of summarization techniques for structured versus unstructured data, such as machine learning, statistical, and natural language processing, have been developed to assist with these tasks:

- **Statistical methods** are commonly used in data summarization to identify patterns and summarize large datasets in a meaningful way[69]. These methods involve using statistical techniques to extract important information from the data, such as central tendency, dispersion, and correlation. One of the most common statistical methods used in data summarization is descriptive statistics, which involves calculating measures such as mean, median, mode, standard deviation, and range to describe the characteristics of the dataset. These measures can help provide insights into the distribution and variation of the data, and are useful for identifying outliers and trends. Another statistical method used in data summarization is regression analysis[23], which involves modeling the relationship between two or more variables and using this model to make predictions. This method is often used in finance and economics to forecast future trends and make investment decisions. Cluster analysis[57] is another statistical method used in data summarization, which involves grouping similar data points together based on their characteristics. This method can help identify patterns and similarities within the data, and is often used in marketing and customer segmentation. Overall, statistical methods are very effective in data summarization as they can help identify patterns and summarize large amounts of data in a meaningful and interpretable way. However, they may require more technical expertise and understanding than linguistic methods, and may not always capture the full complexity of the data.
- **Linguistic methods** Linguistic methods[11] are often used in data summarization to help improve human understanding of the summarized data. These methods involve translating the data into natural language statements or summaries, which can be more easily comprehended by humans. One common linguistic method used in data summarization is fuzzy linguistic summarization [77] This method involves converting data into linguistic terms that are easy to understand, such as "high," "medium," and "low," rather than numerical values. In order to address issues with network traffic flow record summarization tools, Pouzols et al [54] proposed a solution that utilizes fuzzy linguistic summaries based on network traffic. The purpose of this approach is to improve human understanding of the network traffic summaries. The summaries produced by this method are often more readable and informative than traditional

numerical summaries. Another linguistic method used in data summarization is text summarization, which involves summarizing large amounts of text data into shorter, more concise summaries. This method uses natural language processing techniques to extract the most important information from a large body of text, and then presents it in a condensed form. Overall, linguistic methods can be very effective in data summarization as they make the information more accessible and easier to understand, especially for non-experts who may not be familiar with the technical jargon or specific domain terminology.

- **Iterative Compression** Iterative Compression[4] is a data summarization technique that involves compressing a large dataset by reducing the number of rows through grouping similar rows and representing them with a single representative row. The goal is to minimize the size of the dataset while preserving its essential characteristics. IT Compression was proposed by Jagadish et al. [34] for relational databases. The approach tries to compress a relation R by reducing the number of rows by grouping similar rows and representing them by a Representative Row (RR). The IT compression algorithm works by iteratively grouping similar rows until the desired compression level is achieved. During each iteration, the algorithm selects a set of similar rows and replaces them with a single representative row, which is a weighted average of the selected rows. The weights are based on the frequency of occurrence of each attribute value in the selected rows. The IT compression algorithm is particularly useful for summarizing relational databases, as it can be used to compress large tables without losing important information. By reducing the size of the dataset, it can also speed up data processing and analysis. One limitation of IT compression is that it requires some prior knowledge of the data and its structure. Additionally, the compression level may not always be optimal, and some information may be lost in the compression process. Nevertheless, IT compression is a powerful data summarization technique that can be used to efficiently summarize large datasets while preserving their essential characteristics
- **Clustering** Clustering is a data summarization technique that involves grouping similar data points together based on their characteristics. The goal is to identify patterns and similarities within the data, and summarize it in a meaningful and interpretable way [5]. There are various clustering algorithms that can be used for data summarization, such as k-means [52], hierarchical clustering[51] , and density-based clustering [36]These algorithms work by assigning each data point to a cluster based on its similarity to other data points in the same cluster. Clustering is particularly useful for summarizing large datasets, as it can help identify patterns and similarities that may be difficult to discern from the raw data. It can also help identify outliers and anomalies within the data, which can be useful for quality control and anomaly detection. One limitation of clustering is that it may not always capture the full complexity of the data, and may not be suitable for datasets with high levels of noise or overlapping clusters. Nevertheless, clustering is a powerful data summarization technique that can be used to efficiently summarize large datasets and identify meaningful patterns and similarities within the data
- **Stream Clustering** Stream clustering is a data summarization technique that involves grouping data points into clusters in real-time as they arrive in a data stream. The goal is to efficiently summarize the data stream and identify meaningful patterns and trends as they unfold over time. Stream clustering algorithms work by continuously process-

ing incoming data points and updating the clustering model to reflect the changing distribution of the data. Some popular stream clustering algorithms include CluStream [73] ,STREAM [67], and Den Stream[14] One key challenge in stream clustering is the need to balance accuracy with computational efficiency. As data streams can contain large volumes of data and may arrive at high velocities, stream clustering algorithms must be able to process data quickly and efficiently, while still maintaining accurate and meaningful cluster representations. To address this challenge, stream clustering algorithms typically use techniques such as windowing, sampling, and approximation to reduce the computational overhead of processing large data streams. For example, some algorithms may use sliding windows to limit the amount of data that needs to be processed at any given time, while others may use random sampling to reduce the size of the dataset without sacrificing accuracy.

– **Graph summarization** Graph summarization is a data summarization technique [4] that involves representing large graphs in a more compact and interpretable form. Graphs are often used to model complex relationships between entities, such as social networks, biological networks, or transportation networks. However, large graphs can be difficult to visualize and analyze, and may contain redundant or irrelevant information. Graph summarization algorithms work by extracting key features or subgraphs from the original graph that preserve the most important information about the underlying structure and relationships. There are various graph summarization algorithms, such as graph sampling, graph clustering, and graph compression. Graph sampling[31] involves selecting a subset of nodes or edges from the original graph that are representative of the overall structure and relationships. This can be useful for visualizing large graphs or analyzing graph properties that are computationally expensive to compute on the entire graph. Graph clustering[3] involves grouping nodes in the original graph into clusters based on their similarity or connectivity. This can be useful for identifying communities or functional modules within the graph, and for detecting anomalies or outliers. Graph compression involves compressing the original graph by identifying and removing redundant or irrelevant information. This can be useful for storing or transmitting large graphs efficiently, or for summarizing the graph in a more interpretable form. Graph summarization is particularly useful for analyzing large and complex graphs, as it can help identify meaningful patterns and relationships that may be difficult to discern from the raw graph data. By summarizing the graph in a more compact form, graph summarization algorithms can also help reduce computational overhead and storage requirements. In this paper, we will be using graph summarization as a technique for analyzing large and complex graphs. Graphs are often used to model complex relationships between entities, and graph summarization can help us identify meaningful patterns and relationships that may be difficult to discern from the raw graph data. Additionally, by summarizing the graph in a more compact and interpretable form, we can reduce computational overhead and storage requirements. We have chosen this technique because it is particularly useful for our data analysis task and can help us gain insights into the underlying structure and relationships of our graph data.

4. Related Work

In this section, describes and discusses related works on graph summarization approaches, with a particular focus on electronic health record summarization approaches

4.1. Graph summarization approaches

In this section, we will provide an overview of the three main categories of graph summarization: static plain, static labeled, and dynamic graphs.

1. **Static plain graph approach** Most research in static graph summarization focuses on the graph structure without considering side information or labels. In general, the problem of summarization, aggregation, or coarsening of static graphs can be described as simplification-based summarization methods that streamline the original graph by removing less "important" nodes or edges, resulting in a sparsified graph [43]. One example of node simplification-based summarization techniques is OntoVis [64], which is a visual analytical tool that relies on node filtering to understand large, heterogeneous social networks with nodes and links. Toivonen et al. [70] focused on compressing graphs with edge weights and proposed to merge nodes with similar relationships to other entities, which are called structurally equivalent nodes. SPINE, an alternative to CSI [76], sparsifies social networks to keep only the edges that "explain" the information propagation, those that maximize the likelihood of the observed data. In the visualization domain, Dunne and Shneiderman [22] introduced motif simplification to enhance network visualization. They suggested simplifying a graph by extracting its repetitive patterns or motifs and replacing them with higher-level motifs. The simplified graph becomes more interpretable, and the high−level motifs reveal the underlying structure of the graph. In summary, these methods focus on simplifying the graph structure while maintaining its essential properties. By removing redundant or less important nodes and edges, the resulting summary graph is more manageable and easier to analyze while preserving the essential features of the original graph.
2. **Static labeled graph approach** Graph summarization methods that focus on labeled graphs aim to leverage both structural connections and node attributes to produce more informative summaries. These methods face challenges such as the efficient combination of these two different types of data, as well as the selection of meaningful subgraphs or nodes for the summary. One of the most well-known approaches in this category is the frequent-subgraph-based summarization scheme, SUBDUE [19]. It employs a greedy beam search to iteratively replace the most frequent subgraph in a labeled graph. The S−Node representation [59] is another lossless graph compression scheme that optimizes specifically for web graphs. Other approaches, such as SNAP and k-SNAP [69], rely on attribute and relationship-compatibility to group nodes with similar attributes and connections into homogeneous groups. Some recent works have also proposed lossy graph summarization frameworks, such as the collection of d-summaries introduced by Song et al. [38], which group similar entities into supergraphs. Overall, while there have been many advances in labeled graph summarization, it remains an active area of research with many challenges and opportunities for further development.

3. **Dynamic graph approach** Analyzing large and complex data is a challenging task, which becomes even more difficult and time-consuming when dealing with time-evolving networks. The temporal graph mining literature has extensively studied laws and patterns of graph evolution. However, summarization techniques for time-evolving networks have not been studied to the same extent as those for static networks, possibly due to the new challenges introduced by the time dimension. Time granularity, which is often arbitrarily chosen, is a sensitive parameter that can be set to minutes, hours, days, weeks, months, years, or some other unit that makes sense in a given context. TimeCrunch[78] are two methods that succinctly describe a large dynamic graph with a set of important temporal structures. Qu et al.[56] represents a stream of time-ordered interactions, represented as undirected edges between labeled nodes. NetCondense[2] is a node-grouping approach that maintains specific properties of the original time-varying graph, like diffusive properties important in marketing and influence dynamics, governed by its maximum eigenvalue.

4.2. Discussion on graph summarization approaches

To better understand and compare graph summarization approaches, it's important to recognize that the notion of a graph summary is not well−defined. The specific goals and applications of a summary can vary widely, and may include preserving structural patterns, focusing on specific network entities, maintaining query answers, or preserving graph property distributions. Graph summarization approaches can be broadly categorized into three main types: static plain, static labeled, and dynamic graphs. To address the challenges inherent in graph summarization, we propose nine criteria that can be used to describe and compare existing approaches. By considering these criteria in the development and evaluation of graph summarization techniques, we can better understand and overcome the challenges of this important field. We have established nine criteria used to describe and evaluate existing approaches. These criteria are designed to address the specific challenges and goals of graph summarization, and provide a framework for understanding and comparing different approaches.

Challenge 1: How can we generate a summary that integrates data from multiple sources in various formats, such as text, video, and images?

- **Type of input Data (C1)**: this criterion pertains to the input data used in graph summarization approaches. The input data can take on different forms, including (i) structured data such as predefined knowledge models that include existing ontologies and database schema/graphs. (ii)Semi-structured data involves a mix of structured data and free text, such as web pages, Wikipedia sources, dictionaries, and XML documents. (iii) Unstructured data refers to any plain text content, videos, signals, and so on. It's important to consider the nature of the input data when designing and evaluating graph summarization techniques, as the data format can impact the effectiveness and accuracy of the summary.
- **Data type (C2)**: this criterion describe the type of data incorporate (text, xml, numeric, video, image). It's important to consider the type of data being summarized, as this can impact the methods and techniques used in the summarization process. Different data types may require different approaches to summarization in order to effectively capture and convey important information.

– **Representation standard (C3)**: this criterion describes if the approach incorporates standard((i.e. information based standard, document based standard or Hybrid standard) (e.g., Yes or No). The use of a standard can facilitate the comparison and integration of graph summaries from different sources. It's important to consider whether a standard is used in the summarization approach, as this can impact the interoperability and compatibility of the summary with other systems and applications.

Challenge 2: What methods can be developed to generate user-oriented, semantic-based summaries that are tailored to specific information needs and retrieval challenges?

– **Summarization approach (C4)**: this criterion pertains to whether the summarizing approach targets the structure or the content of the graph. Some approaches may prioritize preserving the structural relationships between nodes, while others may focus more on the content and attributes of the nodes (i.e. based structure, dbased content).
– **objective (C5)**: this criterion refers to the specific target or goal of the summarization approach. This could include improving query efficiency, reducing the size of the graph, or identifying influential nodes or relationships. By defining the objective of the summarization approach, it becomes easier to evaluate the effectiveness and relevance of the approach in achieving its intended purpose.
– **Summarization technique (C6)**: this criterion refers to the techniques deployed to summarize data which could be: grouping, compression, analysis, pattern- mining, classification, visualization. The choice of technique used can impact the quality and effectiveness of the resulting summary. By considering the summarization technique employed, we can better understand the strengths and limitations of a particular approach.

Challenge 3: How can we ensure that a summary can effectively analyze and capture the changing nature of real data over time, while still providing a concise and informative representation of the underlying data?

– **Output type (C7):**this criterion concerns type of data displayed in the summary output. which is a combination of: numerical data, textual data, document, graph. By considering the output type, we can better understand the format and presentation of the summary information, which can impact its usefulness and accessibility to end users
– **Context-aware criterion (C8):** this criterion refers to the degree to which a summarization approach takes into account the contextual information surrounding the data being summarized. This can be divided into two types of context-awareness: partial and total. (i)Partial context-awareness refers to the use of concepts related to the dynamic context of the data, such as time, location, and trajectory. (ii) Total context-awareness, on the other hand, refers to the use of both the dynamic context of the data and other contextual information related to the static data. By considering the level of context-awareness, we can better understand the extent to which the summarization approach takes into account the broader context of the data being summarized.
– **User oriented summarization (C9):** pertains to the extent to which a summarization approach is designed with the user in mind. This criterion asks whether the approach

is user-oriented, meaning that it is designed to meet the needs and preferences of the user. By considering the user's needs, the summarization approach can be tailored to present information in a way that is meaningful and useful to the user. This can improve the overall effectiveness and usability of the approach (e.g., yes or No).

Our analysis indicates that the field of graph summarization still faces many challenges. Many existing studies, such as Shen et al. [64], Toivonen et al. [70], Xu et al. [76], do not consider real-world data in their analysis. These studies rely solely on synthetic or simulated data to test their summarization approaches, which may not accurately reflect the complexities of real-world scenarios. Furthermore, most existing systems do not consider the context in which the data is being summarized. They rely solely on time-based properties and do not take into account other important contextual factors that could affect the summary. Consequently, existing systems like Tang et al. [67], and Adhikari et al. [2] are still unable to interpret and reason on the transferred knowledge among real data, which hinders their ability to provide accurate desired results. It is worth noting that existing graph summarization approaches have limited functionality and can only satisfy certain aspects of users' needs. For example, Shi et al. [65] propose a framework for summarizing graphs based on visual representations, while Fan et al. [24] propose a method for summarizing graphs based on user queries. However, none of these approaches provide a comprehensive framework that integrates various functionalities to meet users' diverse needs. Finally, the output type of summarized data is another important consideration. Most existing studies do not propose dedicated tools that make the summary accessible to the user, nor do they provide appropriate perceptions of their needs. Users are increasingly concerned about the security, confidentiality, and accuracy of their data, and existing systems do not adequately address these concerns.

4.3. EHR Summarization Approaches

In the following, we present related works on summarization approaches that are relevant to our domain of application, i.e., Electronic Health Records (EHRs). Discussing the related works on EHR summarization approaches is crucial to validate the motivating scenario presented in Section 2 and further highlight the challenges in this domain. By examining the existing approaches, we can identify the strengths and weaknesses of each and gain a better understanding of the current state-of-the-art in EHR summarization. This knowledge can then be used to guide the development of more effective and efficient approaches.

The EHR concept has appeared since the 1960s [46] and we note that there is no common definition of an EHR until today. The EHR−based application has to be accessible, secure and highly usable. In [27], Gunter and Terry define EHR as a set of clinical and electronic data about a given patient and a population.The World Health Organization (WHO) [10] defines EHR as medical records provided in an EHR-based system aiming at collecting data , storing and manipulating, and providing safe access.

The process of EHR summarization involves creating a summary that contains the most relevant information from the original content. Previous research in this area has focused on text summarization, with the aim of providing useful information for healthcare professionals by automatically compressing a given text [47]. The type of summary required may vary depending on the clinician's needs, but in general, it should cover

as much of the medical content as possible, while preserving the overall topical organization of the original text. This review specifically focuses on approaches based on multi-document extractive summarization, which involve producing a summary of multiple documents about the same patient. These summarization approaches are typically focused on extracting clinical variables and visualizing structured and unstructured data [53], in order to provide an overview of the patient's entire medical record. We will discuss and analyze 38 research papers on EHR summarization, which are categorized into four types of EHR text summarization:

1. **Extractive Summaries** EHR summarization typically involves selecting a subset of information from the original content, with text summarization being the primary focus of existing studies [47]. The aim is to provide a compressed version of the text that is relevant to the needs of clinicians. Generic summaries covering the medical content in multiple documents must maintain the general topical organization of the original text. This method synthesizes patient records by displaying the summary in user-friendly interfaces. Studies have explored supervised approaches to extractive summarization, such as [45], which trained a transformer-based neural model using International Classification of Diseases (ICD) codes for specific diagnoses. Radiologists evaluated the approach and found that supervised models generate better summaries than unsupervised approaches. The model aims to include accurate components of EHR data, such as structured data, sentence-level clinical aspects, and structures of clinical records. Authors provide a clinical data processing pipeline based on NLP and the use of concept recognition and relation detection. Other studies have explored the use of NLP to customize user views, such as [41], which uses MedLEE NLP engine to handle modifiers. Some studies have also explored generating meaningful topic summaries from structured clinical data, such as [26], which learns the correspondences between structured data and clinical note topics using existing summaries written by clinicians. Approaches have also been proposed for synthesizing clinical data, such as the SIM card-based system in [1], which displays synthesized clinical data on mobile phones using custom-developed software. [68] proposed a summarization approach to classify patients with and without diabetes, evaluating the approach using traditional classification methods and machine-learning techniques. Another study [66] focused on metastases information extraction from pathology reports of metastatic lung cancer. In [48], authors propose a Bayesian summarization method for summarizing biomedical text documents, involving mapping the input text to the Unified Medical Language System (UMLS) terminology and selecting relevant ones to use as classification features. Finally, [53] proposes UPhenome, an approach based on graphical models and large scale probabilistic phenotyping to model diseases and patient characteristics and generate summarized clinical data. [30] proposes a real-time summarization approach by aggregating clinical data from heterogeneous health care systems using HL7 messages and a distributed architecture.
2. **Abstractive summaries** Abstractive summarization techniques are a type of text summarization approach that goes beyond simply extracting relevant phrases or sentences from the original text. Instead, they generate new text by synthesizing the most important information from the source material. This can provide additional context and insight beyond what is available in the original text[55]. In the medical domain, researchers have explored a variety of abstractive summarization techniques.

For example, [13] and [9] proposed a method called Timeline, which involves clinicians in coding rules to generate abstractive summaries. Another approach, called AdaptEHR, was developed by [13] to infer rules and relationships automatically from ontologies and graphical models. In [63], a hybrid abstractive-extractive summarization approach was proposed. This method aims to perform semantic, temporal, and contextual abstraction using a domain-specific ontology to generate abstractions. Additionally, [33] and [72] explored a graphical approach to summarizing clinical data by generating new text. Overall, these studies demonstrate the potential for abstractive summarization techniques to provide more comprehensive summaries of medical records.

3. **Indicative Summaries** Indicative summarization is an approach that extracts significant terms from the original text and highlights the main parts. This technique is used in conjunction with EHR to indirectly integrate the extractive summarization process. However, there are limited studies in the literature concerning indicative summarization. Rogers and colleagues proposed a new approach to summarize and graphically visualize the EHR, including indicative summaries. Their approach involved creating task-based evaluation summarizers, which extracted the most relevant information for specific clinical tasks. In another study, Clayton and colleagues evaluated how and when clinicians in an ambulatory setting would enter data directly into an EHR. They found that physicians entered more information when the patient's problem was acute, complex, or unfamiliar [18]. This study did not directly focus on indicative summarization, but it provided insight into how physicians interact with EHR and how information is entered into the system. [61] proposed a new approach to summurize and graphically visualize the EHR. [61] proposed a task-based evaluation summarizer. In [18] authors evaluated how and when clinicians in an ambulatory setting will enter data directly into an EHR.
4. **Informative Summaries** The informative summarization approach is distinct from the other approaches in that it aims to replace the original set of raw data rather than simply provide an abstract of it. This approach is designed to create summaries that can be used independently of the EHR. Several studies have proposed methods for informative summarization. For example, Matheny et al. [44] developed a new model for summarizing structured clinical data, such as administrative, computerized provider order entry, and laboratory test data. Their model was used to detect risks by predicting two severity levels of in-hospital Acute Kidney Injury (AKI). Visualization-based summarization approaches have also been proposed in the literature, such as those by Bade et al. [8], Wang et al. [74], and Borland et al. [12]. These methods aim to create summaries by visualizing the data in a more understandable and concise manner. RDF-based summarization approaches have also been proposed, such as Carenini et al. [15]. These methods use RDF (Resource Description Framework) to represent and summarize text data. Many research groups, such as the NUCRSS [61], have proposed clinical data summarization systems based on text input data. These systems aim to reduce the volume of data and make it more manageable for clinicians. Various frameworks have also been proposed for text summarization, such as those by Liu et al. [40], Wright et al. [75], and Radev et al. [58]. In addition, methods for generating new stories and scientific articles to summarize unstructured texts have been proposed, such as those by Nenkova et al. [50], Lukasik et al. [42], Liu et al. [40], and Reeve et al. [60].

4.4. Discussion on EHR summarization approaches

To address the challenges related to EHR summarization and to facilitate a comparison of existing studies on clinical data summarization, the criteria outlined in the previous section were utilized to characterize and compare different methods of clinical data summarization in accordance with the challenges identified in the introduction and we mentionned also other criterion in the same context such as Representation standard (C3), this criterion indicates if the approach incorporates standard((i.e. information based standard, document based standard or Hybrid standard) (e.g., Yes or No). This approach enabled a comprehensive evaluation of various techniques used for EHR summarization, including the extent to which they address the challenges associated with this task. By employing these criteria, researchers can gain insights into the strengths and limitations of different summarization approaches and can make informed decisions about which method is best suited to their specific use case. Additionally, this approach facilitates a comparative analysis of different studies on EHR summarization, providing a better understanding of the current state of the field and identifying areas for future research.

According to the analysis presented in Table 3, the existing approaches for clinical data summarization can be categorized into two groups: those that rely on structured data and those that use unstructured data. The studies [49] fall under the category of approaches that are based on structured data. On the other hand, the studies [15] belong to the group of approaches that rely on unstructured data. It is important to note that none of the approaches analyzed in this study utilize both structured and unstructured data to construct the summary. Another important aspect of the analyzed studies is the type of output provided by the clinical data summarization systems. The studies reviewed in this paper propose either document-based systems [26] or graph-based systems [12]. However, none of these studies offer dedicated tools that facilitate user access to the summary or provide them with appropriate perceptions of their needs. Users are increasingly concerned about the security, confidentiality, understanding, accuracy, and completeness of their data. Therefore, it is essential to ensure that clinical data summarization systems provide users with the necessary means to ensure these aspects. An intuitive and user-friendly graphical user interface (GUI) would significantly benefit clinical data summarization systems. Also, we highlight that none of the studies surveyed are user-oriented and able to satisfy the diverse needs of users. Our analysis, presented in Table 2, reveals that the evolution of data summarization is still an ongoing challenge. Most of the existing studies fail to consider the contextual information of the data in their analysis and do not take into account the context when creating the summary. Instead, they rely solely on the time property, except for a few studies that consider time in their analysis . As a result, current systems are still unable to contextually interpret and reason on the transferred knowledge among real data, and thus cannot synthesize data to provide accurate desired results. It is noteworthy that all existing systems focus on only one objective, while none of them provide multiple functionalities within the same framework, despite their importance in supporting users' preferences to find data according to various needs. Therefore, it is necessary that all objectives should be integrated into a summarization-based system. In addition, our analysis shows that most of the studied approaches are extractive-based, including [45],[39],[40], and [26]. However, three of the studies, namely [13], and [9], are abstractive-based. Furthermore, five studies, including [13], adopt both extractive and abstractive−based approaches. In summary, existing systems tend to focus on only

one objective and adopt extractive-based approaches. Nonetheless, it is important to consider multiple functionalities within the same framework and integrate both extractive and abstractive−based approaches in the summarization-based system. Based on our comparative study, we have identified four main limitations in clinical data summarization:

- **Lack of access to and collection of data from Medical Devices:** Due to the heterogeneity of applications, it is critical to synthesize health data in order to provide a relevant, comprehensive, and understandable view of the patient's history to effectively help clinical diagnostics [37].
- **Lack of semantic interoperability:** Applications generate a huge amount of heterogeneous data, which makes it difficult to synthesize knowledge and communicate between clinical applications to provide efficient results [71].
- **Lack of linking Data and medical devices to their contexts:** It is important to describe the data and device context in order to identify its capacity and reliability to ensure the consistency of the gathered data and to easily repair it when necessary [20].
- **Lack of user-centered summary design:** Existing systems are unable to generate adaptive summaries that adjust based on clinician preferences and needs, leading to increased cognitive workload for clinicians [79] It is nearly impossible to provide interactive and personalized summaries, which can result in reduced efficiency and effectiveness in clinical decision−making.

Table 2. Qualitative Comparative study of static, static labeled dynamic plain Graph summary

()2-10	**Challenge 1**			**Challenge 2**				**Challenge 3**	
Existing study/Criterion	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Category 1: Dynamic graph									
Adhikariet al. 2017	Structured	Weighted,Directed	Yes	Structure	Influence	Grouping	Supergraph	Time	Yes
Tan et al. 2016	Structured	Weighted,Directed,	Yes	Structure	Query	Grouping	Supergraph	Time	Yes
Qu et al. 2014	Structured	Unweighted, Undirected	yes	Structure	Influence	Influence	Subgraph	Time	Yes
Category 2: Static graph									
Zhu et al.2016	Structured	Weighted, Undirected	No	Structure	Visualization	Grouping	Supergraph	No	No
Riondato et al. 2014	Structured	Weighted, Undirected	No	Structure	Query	Grouping	Supergraph	No	No
Koutra et al. 2014	Structured	Unweighted, Directed	No	Structure	Visualization	Compression	Graph	No	No
Dunne et al. 2013	Structured	Unweighted, Directed	No	Structure	Visualization	Grouping	Supergraph	No	No
Mathioudakis et al. 2011	Structured	Weighted, Directed	No	Structure	Influence	Influence	Sparsified graph	No	No
Category 3: Static labeled graph									
Song et al. 2016	Structured	Unweighted	No	Structure	Query	Grouping	Supergraph	No	No
Shi et al. 2015	Structured	Weighted, Directed	No	Structure	Influence	Influence	Supergraph	No	No
khan et al. 2014	Structured	Unweighted Directed	No	Structure	Compression	Compression	Supergraph	No	No
Hassnlou et al 2013	Structured	Weighted directed	No	Structure	Grouping	Compression	Supergraph	No	No
Fan et al. 2012	Structured	unweighted, Directed	No	Structure	Query	Grouping	Supergraph	No	No
Zhang et al. 2010	Structured	Unweighted, Undirected	No	Structure	Patterns	Grouping	Supergraph	No	No
Proposed approach	Structured, Unstructured	Labeled, Directed	Yes	Structure, Content	Summarization	Grouping, Aggregation, Mathematical operations, Compression	Supergraph, Subgraph, Graph	Yes	Yes

Table 3. Summary of retrieved studies on electronic health record and medical data summarization

	Challenge 1			**Challenge 2**			**Challenge 3**		
Existing study/Criterion	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Denis Jered et al. 2020	Document	Unstructured	No	Structure	Classification	Extractive	text	No	No
Liang et al. 2019	Text	Unstructured	No	Structure	Classification	Extractive	Text	No	No
PDurga et al.2018	Text	Unstructured	No	Structure	Classification	Extractive	Text	No	No
Jen et al. 2018	Document	Unstructured	No	Structure	Classification Filtering	Extractive	Text	Time	No
Soysal et al. 2017	Text	Unstructured	No	Structure	Visualization	Extractive	Text	No	No
Razavian et al.2015	Text	Unstructured	No	Structure	Analysis	Extractive	Text	No	No
Borland. 2014	Text	Unstructured	No	Structure	Visualization	Extractive	Text	No	No
Fei et al. 2013	Numerical	Unstructured	No	Structure	visualization	Extractive	Numerical	No	No
Klann et al. 2013	Document	Unstructured	No	Structure	Classification	Extractive	Text	No	No
Roque et al. 2010	Document	Unstructured	No	Structure	Visualization	Extractive	Text	No	No
Barakat et al. 2010	Text	Unstructured	No	Structure	Analysis	Abstractive	Text	No	No
Savova et al. 2010	Text	Unstructured	No	Structure	Analysis	Extractive	Text	No	No
Krummenacher et al. 2009	XML	Structured	No	Structure	visualization	Extractive	Tuple	No	No
Kumar and al. 2008	Text	Unstructured	No	Structure	Extraction	Extractive	Text	No	No
Huang et al. 2007	Text	Unstructured	No	Structure	Extraction	Extractive	Text	No	No

5. Contribution

The main contribution relies developing a schema-driven approach for handling heterogeneous data sources by modeling and summarizing them within labeled data graphs. The resulting graph summaries are then visualized to meet the specific needs of each user. In essence, the approach relies on using data graphs and a schema to drive the data modeling and summarization process. Our approach aims to provide a comprehensive and personalized model capable of summarizing both the structure and content of data from databases, devices, sensors.etc. We took into consideration the user needs. In order to achieve our goal, our framework architecture includes four main modules, as illustrated in figure 2. First, the Data Collection module is responsible for collecting data from various sources. Second, the Schema Generation module is used to generate a schema based on the collected data, which enables us to create a labeled data graph. Third, the Summarization module summarizes the data graph based on user-submitted questions, and fourth, the Visualization module provides a personalized visualization model to represent the summarized data graph for each user need. Our approach addresses the limitations identified in the comparative study by providing a user-centered, schema-driven approach that can effectively model and summarize data from heterogeneous sources. By providing a personalized visualization model, our approach aims to reduce cognitive barriers related to the complexity of information and its interpretation, ultimately supporting clinical decision-making.

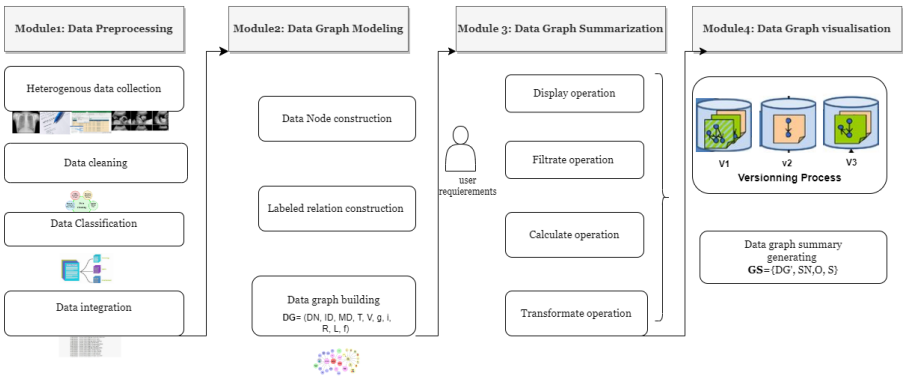

Fig. 2. Architecture of our proposed system

A) **Data Pre-Processing service**: consists of processing and indexing data in order to summarize them. Every incoming data is processed and transformed according to two-steps: data cleaning and data integration. This module is composed of:

1) Data collection: this module is responsible for collecting data in various formats such as pdf documents, images, videos, and numeric data.
2) Data Cleaning: this module involves transforming raw data into an understandable format by extracting data from multiple and heterogeneous sources.

3) Data classification: this module classifies the different data into associated types. To classify documents from a heterogeneous corpus, this module involves text analysis, keyword extraction, and natural language processing.
4) Data Integration: this module integrates different normalized data into a generic framework that supports the direct generation of data in a common format. The objective is to create a unified and consistent data structure that can be used for analysis and modeling purposes.

B) **Data Graph Modeling**: This module is responsible for defining the data graph proposed by our system. The data graph is composed of data nodes and relations between them. The purpose of this graph is to represent the heterogeneous data in a common format, making it easier to process and analyze. By representing data in a structured graph, we can identify patterns, relationships, and dependencies between data nodes, which can be used to generate insights and support decision-making processes. The data graph will be defined in the next section, and it will serve as the backbone of our system for data processing and analysis.
C) **Data Graph summarization**: this module defines the data processing steps that are involved in generating a data summarization model-based graph, which is the core module of our framework. The purpose of this module is to transform input data into a summarized output. It aims at summarizing data using a driven schema approach based on both structure and content. This involves analyzing the data graph schema and extracting relevant information based on user requirements, which are represented using different functions. To achieve this, the module has multiple microservices that operate based on the user needs and requirements. The data summarization model-based graph will be further detailed in section 7
D) **Data Graph visualization**: this module is responsible for the visual representation of data. Its main objective is to provide visual and interactive visualization of the summarized data to help users rapidly find insights in data. It includes interactive techniques to graphically represent the summary, such as charts, graphs, tables, and dashboards. These visual representations provide an intuitive and easy-to-understand view of the summarized data, allowing users to explore and analyze it in a more efficient way. This module generates a graph summary which is a visual representation of the summarized data using the data graph schema.

6. Data Graph Modeling

In this section, we focus on the data graph modeling module, which is responsible for creating a graph-based data model of the input data. The module performs an aggregation process on the transformed input data and generates an aggregated value.

The data graph model is built iteratively, starting from a root node that represents the whole data set. Each aggregated item in the graph consists of one or more children, which can be either original data items (leaves) or aggregated items (nodes). This hierarchy of nodes and leaves allows for a structured representation of the data, which makes it easier to analyze and summarize. To achieve this, we introduce a new Data Graph Model (DGM), which serves as a common synthesis of a large amount of data. The primary goal of the DGM is to facilitate and perform the summarization process by providing a structured representation of the input data.

6.1. Data Graph Definition

The data graph (DG) is a graph-based representation of the input data. It consists of Data Nodes (DN), also known as data entities, which model heterogeneous data such as text, images, videos, and numerical data. These DNs are the building blocks of the graph and serve as the vertices in the graph. In addition to the DNs, the DG also contains relations between them. These relations are the edges in the graph, which connect the DNs to each other. Each relation has a label that defines the nature of the connection between the DNs.

6.2. Data Graph Formalism

Definition 1: Data Node (DN) A Data Node (DN) is a fundamental component of the Data Graph, representing the information contained within a data structure. Each DN holds a value of structured or unstructured data and is associated with a single parent node. The DN is defined by its identification, name, and metadata, which describe its type, value, and acquisition time. It also has a set of attributes, each with a corresponding value and data type. These attributes define the specific properties of the DN and provide additional information about its contents.

Definition 2: Data Graph (DG) A Data Graph (DG) is a representation of important structured and unstructured data within a domain. The purpose of defining the DG is to create an efficient representation of domain data and the relationships between them. Formally, the DG is represented as:

$$\text{DG= (DN, ID, MD, T, V, g, i, R, L, f)}$$

where

- DN: is a set of Data Nodes
- ID : is a set of identifier of Data Node
- MD: is set of attributes (e.g DocName, Date...)
- T: is a set of types $\{String, number, Boolean\} \cup$\{Array,image, son, video\}
- V: is a set of values
- g: DN $\rightarrow$ (MD, T, V)
- i: DN $\rightarrow$ ID
- R: set of relationships: R $\subseteq$DN $*$ DN
- L: set of terms
- f: R$\rightarrow$ L

Algorithm 1: Data Graph Generation

```
   Input  : Heterogenous Data
   Output: DG
 1 Initialize an empty graph DG
 2 for i from 1 to n (n is the total number of data nodes in DN) do
 3 |   Add a node to graph G with identifier i(DN[i])
 4 end
 5 for i from 1 to n do
 6 |   for j From 1 to n do
 7 |   |   if there exists a relationship r in R such that DN[i] is either the source or the destination of r then
 8 |   |   |   Add an edge to graph DG between nodes i(DN[i]) and j(DN[j])
 9 |   |   |   If
10 |   |   end
11 |   end
12 end
13 for i from 1 to n do
14 |   Add a set of attributes to node i(DN[i]) based on the values in MD[i]
15 end
16 for i from 1 to n do
17 |   Add a set of values to the node i(DN[i]) based on the values in V[i]
18 end
19 for i from 1 to n do
20 |   Add a set of types to node i(DN[i]) based on the values in T[i]
21 end
22
23 for  i from 1 to n do
24 |   Add a tuple (MD[i], T[i], V[i]) to the mapping g for node i(DN[i])
25 end
26 for each relationship r in R do
27 |   Add a term l to the set L based on the function f
28 end
29 Return the DG
30
```

This pseudo code describes an algorithm 1 that generates a data graph. The algorithm first initializes an empty graph DG. Then, for each index i from 1 to n (number of data nodes), it adds a node to graph DG with identifier i(DN[i]). Next, for each pair of indices i and j from 1 to n, it checks if there exists a relationship r in R such that DN[i] is either the source or the destination of r. If there is such a relationship, it adds an edge to graph G between nodes i(DN[i]) and j(DN[j]). After that, for each index i from 1 to n, it adds a set of attributes to node i(DN[i]) based on the values in MD[i], a set of values based on V[i], and a set of types based on T[i]. Then, for each index i from 1 to n, it adds a tuple (MD[i], T[i], V[i]) based on the function g for node i(DN[i]). Finally, for each relationship r in R, it adds a term l to the set L based on the function f. The algorithm returns the graph DG as the data graph.

This data includes various resources that provide information about the patient's personal details, medical history, and current health status. Specifically, the data sources consist of medical tests, prescriptions, scans, and radiographs that are associated with the patient. The DG generated is a graphical representation of this data, where nodes represent the different data sources and edges represent the relationships or connections between them.

7. Data Graph summarization

Graph summarization is the process of creating a Graph Summary (GS) from a given DG. The GS captures the essential features of the DG, while reducing its complexity and size. The process of graph summarization involves selecting a subset of the vertices and edges

from the DG to be included in the GS, while still maintaining the overall structure and connectivity of the original graph.

7.1. Data Graph summary Definition

The Graph Summary (GS) contains Summary Nodes (SN) and Operational Relationships, which are represented as functions called Operation (O). These O correspond to various user needs, such as display, filtering, transformation, and calculation. The SN in the GS represent a subset of the nodes in the original DG that capture the most relevant information. The Operations, define how the SN can be generated to satisfy user needs. By using the GS and its associated operations, users can interact with the data in a more efficient and effective way. The GS reduces the complexity of the original DG while still preserving the important information, and the functions provide a flexible and customized way to perform various operations on the summary nodes.

7.2. Data Graph summary formalism

Definition 3: Summary Node (SN) A Summary Node (SN) is a representation of a subset of nodes in the DG that captures the most relevant information. The SN is a node in the GS that summarizes a group of nodes in the original DG and can be generated using various operations to satisfy user needs. The SN is defined identically to DN.

Definition 4: Graph Summary (GS) A Graph Summary (GS) is a condensed representation of a larger Data Graph (DG) that contains Summary Nodes (SN) and Operational (O) in the form of functions. The SNs represent a subset of the nodes in the original DG that capture the most relevant information, while the (O) define how the SNs can be generated to satisfy user needs, such as display, filtering, transformation, and calculation. Formally a GS is defined as follow:

$$SG \doteq (DG', SN, O, S)$$

Where

- DG: $DG' \subseteq DG' * DG$ (DG' is a part of DG)
- SN: is a set of summary nodes, SN is defined identically to DN
- O: is a set of synthesis operations (max, min, avg, display, filtrate, transformate, calculate)
- S: $O \rightarrow DN * SN \cup DN$

The formalism for the Graph Summary (GS) involves defining the SG as a tuple containing a subset of nodes in the original Data Graph (DG'), Summary Nodes (SN), Operational Relationships (O), and a mapping function (S) that defines how the SNs can be generated to satisfy user needs. The subset of nodes in the original DG, denoted as DG', is a part of the original DG and serves as the basis for the summary. The SNs in the GS represent a subset of the nodes in the original DG that capture the most relevant information. These SNs can be generated using various functions represented by the operational relationships (O), such as display, filtering, transformation, and calculation. The mapping function (S) maps these operational relationships to the set of nodes in the original DG

and the set of SNs in the GS. In summary, the Graph Summary formalism defines a condensed representation of a larger Data Graph that captures the most relevant information using a subset of data nodes and operational relationships. The mapping function allows for flexible and customized ways to generate Summary Nodes to satisfy user needs.

Definition 5: Operation (O) An Operation (O) is a function in the GS defining how SN can be generated to satisfy user needs, such as displaying, filtering, transforming, and calculating. The O $\rightarrow$ DN*SN mapping function links the operational relationships to the set of DN in the original DG and the set of SNs in the GS.

Definition 6: Display Operation The Display Operation, denoted by Display(nodetype), is used in constructing the GS based on the projection operator. Its main objective is to remove a specific DN from a common relation in the data graph. The nodetype parameter specifies the type of node to be projected or displayed in the summary. In other words, the display operation is used to generate a SN that captures only the relevant information from a larger set of data nodes.

Algorithm 2: Display Operation (Same NodeType)

Input : DG
Output: GS
Create an empty set of Summary Nodes SN and an empty set of Operations O
Identify important nodes in DG that have a large number of connections or are in the center of the graph
Group the identified nodes into a node grouping DG'(DN): $\text{dn}_1, dn_2, \ldots, dn_m$
foreach *DG'i in DG'* **do**
    Create a Summary Node SN(GSj) and assign it an unique identifier j
end
foreach *DNj in DN(DGi)* **do**
    assign it to SN(GSj) r
end
foreach *dni, dnj in DG* **do**
    if *i=j and sni, snj belong to both DG'i and SGj* **then**
        |
    end
    Then connect them with an edge in GS
end
Return Gs= (DG', SN, O, S)

The following pseudocode outlines the steps for creating a graph summary GS from an input DG. Firstly, an empty set of Summary Nodes SN and Operations O is created. Then, significant nodes in the input DG are identified based on their number of connections or centrality and grouped into a node grouping DG'(DN).For each DG'i in DG', a Summary Node SN(GSj) is created and assigned a unique identifier j. Each DNj in DN(DGi) is then assigned to SN(GSj). For each dni, dnj in DG, if i=j and sni, snj belong to both DG'i and SGj, they are connected with an edge in GS.

Finally, the algorithm defines Operations O that can be used to generate the Summary Nodes SN to satisfy user needs, such as display, filtering, transformation, and calculation.

This algorithm 3 takes as input a graph DG and outputs a graph GS. It first defines the join operation as and then iterates over each DNj in DN(DGi) to select data nodes and metadata based on certain conditions. For each selected data node dn1, it selects metadata md1 such that MD is either in the data nodes DN= dn1, dn2, ..., dnm or MD is the metadata of dn1 or MD is in nodetype and equals acquisition time md1, md2, ..., mdn in DG. The algorithm then selects the relation r1 such that r1 is in DR or r1 is in R(r2$\in$ o

Algorithm 3: Display Operation (Multiple Node Type)

Input : DG
Output: GS
foreach *DNj in DN(DGi)* **do**
  Define the join operation as ⋈:
  DG x DG x cond → G's(dn1, dn2, condj) → dn1, dn2, condj
  Select data nodes based on the condition Let dn1 be the selected data node where $dn_1 \in$ data nodes DN and $dn_1, dn_2, dn_3, .., dn_n (dn_2 \in$ DN $dn_1, dn_2, \ldots, dn_m)$
  Select meta data based on the condition:
  Let md1 be the selected meta data where:
  MD ∈ data nodes
  DN= dn1,dn2,dn3,...,dnn ∨ (md1 ∈ DN= dn1,dn2,...,dnm)
  Let MD be the data node where MD is the meta data of dn1
  or ($md_2 \in$ nodetype | MD = acquisition time md1, $md_2, \ldots, md_n \in$ DG)
  Let r1 be the selected relation where r ∈ DR ∨ (n2 ∈ O r1 ∈ $R(r2 \in$ o | r = r1,r2,...,rn∈ $DG) \wedge$, o, G's
end
Return GS=(DG', SN, O,S)

| r = r1,r2,...,rn∈ DG), and o and G's are determined.
Finally, the algorithm returns GS= (DG', SN, O, S) where DG' contains tuples (dn1, dn2, condj, md1, r1), SN contains tuples (dn1, dn2, condj), O contains tuples (r1, o), and S contains tuples (md1, GS').

Definition 7: Filtering Operation The filtering operation is an operator that selects a subset of nodes from a DG based on a selection predicate or criterion, which produces a GS. The filtering operation uses a selection predicate to determine which nodes in the DG should be included in the GS. The predicate, or selection criterion, can take on various forms, including a combination of node type, attribute, and relation. In other words, the valid selection criteria can be represented as a condition that specifies the node types, attributes, and relations that should be included in the GS.

The Filtering Operation algorithm takes as input a data graph (DG) and outputs a graph summary (GS) by selecting relevant attributes for each data node. The algorithm starts by initializing an empty set of selected attributes, MD Set. For each data node DN in DG, if DN is in the set of multiple data nodes (MD), then the algorithm selects metadata nodes (mdi) that satisfy the set of valid selection criteria (cond) and adds them to the set of selected MD Set. If G's is a supernode, the algorithm selects metadata nodes from the set of selected attributes (MD Set) and adds them to the summary graph GS.

Definition 8: Transformate Operation A Transformation Function consists of transforming numerical data into graphical or tabular format. This type of transformation function involve a variety of mathematical or statistical operations allowing to manipulate and analyze numerical data in meaningful ways.

Algorithm 5 represents the transformation operation. The input to this algorithm is a data node, and the output is a supernode, which can be in the form of a numerical node, a graphical representation, or a table. The local variable 'Operator(o)' is either 'calculmax', 'calculmin', 'calculavg', or a statistical operator. If the numerical node is one of the numeric nodes dn1, dn2, dn3, ..., dnn in the data graph DG, and the operator is 'calculmax', the algorithm calculates the maximum value of the nodes and creates a new supernode containing the numerical node and the maximum value. The algorithm then returns the supernode. If the numerical node is one of the numeric nodes dn1, dn2, dn3, ..., dnn in the data graph DG, and the operator is 'calculmin', the algorithm calculates the minimum value of the nodes and creates a new supernode containing the numerical

Algorithm 4: Filtering Operation (For Same or Multiple Data Node)

Input : DG
Output: GS
Initialize an empty set of selected attributes, MD_set. **foreach** *Data Node DN ∈ DG* **do**
  if *($DN \in MDN$)* **then**
    foreach *mdi* $\in MD(dn)$ **do**
      if *((mdi) ∈ the set of valid selection criteria cond)* **then**
        Add mdi to the set of selected MD_set
    end
  end
  if *(G's is a supernode)* **then**
    foreach *metadata node md ∈ G's* **do**
    end
    if *(mdi ∈ MD(DN))* **then**
      Add mdi to the set of selected attributes, MD_set
    foreach *dN in DG* **do**
      foreach *mdi ∈ MD(DN) such that MDi is MDi-compatible* **do**
        // MDi-compatible refers to a grouping ϕ that satisfies the condition that for every data node dni in DN, if ϕ(dni) = ϕ(dnj,G's)
        if *($MDi \in thesetofselectedattributes$)* **then**
        Add the corresponding metadata node mdi(G's) to the summary graph GS.
      end
    end
  foreach *mdi∈a in MD(dN) such that mdi ∈ set of selected MD* **do**
    Add the corresponding metadata node mdi(G's) to the GS
  end

Algorithm 5: Transformation operation

Input : DN=numerical node
Output: SN=(numerical node, graphical, or table),GS
Local Variables: $Operator(o) = calculmax, calculmin, orcalculavg$
if *(DN ∈ numeric nodes dn1, dn2, dn3, ..., dnn in DG o≐ calculmax)* **then**
  Calculate the maximum value of the nodes: maxVal ≐ max(dn1, dn2, dn3, ..., dnn)
  Create a new supernode sn ⊆
  SN(Gs) ≐ DN(numericnode; dn1, dn2, dn3, ..., dnn) ⊆ DG
  o ≐ maxVal
Return Gs
if *(DN ∈ of numeric nodes dn1, dn2, dn3, ..., dnn ⊆ DG and o ≐ calculmin)* **then**
Calculate the minimum value of the nodes: minval≐ min(dn1, dn2, dn3, ..., dnn)
Create a new supernode Sn ∈ Gs
SN(Gs) ≐ DN(numericnode; dn1, dn2, dn3, ..., dnn) DG
o≐ minVal
Return G's
if *(DN of numeric nodes dn1, dn2, dn3,..., dnn in DG and o is calculavg)* **then**
  Calculate the average value of the nodes: avgVal ≐ average(dn1, dn2, dn3, ..., dnn) Create a new supernode sn ∈ Gs, SN(Gs) ≐ DN(dn1, dn2, dn3, ..., dnn) of DG and the average value avgVal
Return Gs **if** *(DN ⊆ set of numeric nodes dn1, dn2, dn3, ..., dnn ∈ DG and o is a statistic operator)* **then**
  Calculate the statistical data of the nodes (e.g. standard deviation, variance
  Create a new supernode sn ∈ SN(G's) ≐ table(G's) of DG and the calculated statistical data
Return Gs

node and the minimum value. The algorithm then returns the supernode. If the numerical node is one of the numeric nodes dn1, dn2, dn3, ..., dnn in the data graph DG, and the operator is 'calculavg', the algorithm calculates the average value of the nodes and creates a new supernode containing the numerical node and the average value. The algorithm then returns the supernode. If the numerical node is a set of numeric nodes dn1, dn2, dn3, ..., dnn in the data graph DG, and the operator is a statistical operator, the algorithm calculates the statistical data of the nodes (e.g., standard deviation, variance) and creates a new supernode containing the table of the data graph and the calculated statistical data. The algorithm then returns the supernode. If the numerical node is a single numerical node, and the operator is a statistical operator, the algorithm calculates the statistical data of the node (e.g., standard deviation, variance) and creates a new supernode containing the graphical representation of the data graph and the calculated statistical data. The algorithm then returns the supernode.

Definition 9: Calculation operation The Calculation operation is intended to perform mathematical computations on one or multiple DN. It involves various mathematical operations like addition, subtraction, multiplication, division, exponent, and remainder. Two inputs are needed for each operation, and the requirements of inputs vary depending on the type of mathematical operation being performed. For instance, Add, Subtract, Multiply, Divide, and Remainder nodes can accept numeric values or certain time-based values as inputs, while Exponent operations only accept numeric values. Only one output is produced by each operation, representing the outcome of the mathematical calculation performed on the input data node. It is important to note that these calculations can only be carried out on numerical data nodes

Algorithm 6: Calculation Operation (Numerical Node)

Input : DN
Output: SN
Local Variables: $o, FirstVal, SecondVal$
1 SN $= \varnothing$
2 **if** *(DN.FirstVal* $\in \mathbb{R}$ *and DN.SecondVal* $\in \mathbb{R}$*)* **then**
3 **if** *(*$O == +$*)* **then**
4 $SN =$ DN.FirstVal $+$ DN.SecondVal
5 **else if** *(*$O == -$*)* **then**
6 $SN =$ DN.FirstVal $-$ DN.SecondVal
7 **else if** *(*$O == *$*)* **then**
8 $SN =$ DN.FirstVal $*$ DN.SecondVal
9 **else if** *(*$O == /$*)* **then**
10 $SN =$ DN.FirstVal/DN.SecondVal
11 **else if** *(*$O == \%$*)* **then**
12 $SN =$ DN.FirstVal%DN.SecondVal
13 **else if** *(*$O ==$*)* **then**
14 $SN =$ DN.FirstVal$^{\text{DN.SecondVal}}$
 `// If the operator is not any of the above, return an error message indicating that the operator is not supported.`
15 Return SN

Algorithm 6 describes the Calculation Operation algorithm for numerical nodes. It takes a numerical DN as input and produces a supernode SN as output. The algorithm is based on various mathematical operations such as addition, subtraction, multiplication, division, exponent, and remainder. These operations can only be performed on numerical data nodes. The algorithm begins by initializing the supernode SN to an empty set. It then checks if both FirstVal and SecondVal of the input DN are real numbers. If so, the algorithm proceeds to check the operator O. If the operator is addition (+), then the algorithm

calculates the sum of the FirstVal and SecondVal and stores it in SN. Similarly, if the operator is subtraction, multiplication, division, exponentiation, or remainder, the algorithm performs the corresponding mathematical operation on the FirstVal and SecondVal and stores the result in SN. If the operator is not any of the above, the algorithm returns an error message indicating that the operator is not supported. Finally, the algorithm returns the supernode SN as output.

8. Summary Versioning Process

Definition 9: Summary Versioning Summary Versioning is a process of creating and maintaining different versions or snapshots of a graph Summary at different points in time, allowing users to track and analyze changes to the DG over time. This process can help to identify trends, patterns, and inconsistencies in the data, as well as to undo any mistakes made during the editing process. Formally, a history H is a sequence of one or more quadruplets representing versions of the graph summary, defined as:

V_Gs= (DG, O, V, tn)

where :

- DG: is the original Data Graph input
- O: is the Operator used for the summarization process
- V: is the sequence of graph summary versions
- tn: is the history of the last version of the graph summary at time t, where t is any point in the sequence of time intervals of the history. This means that for each new version of the GS, a new timestamp is added to the history. The history provides a record of all the changes made to the GS over time, allowing for a better understanding of its evolution. The VGs component captures the differences between the different versions of the GS. The versioning process in our approach involves keeping track of all the changes made to the graph summary over time, and representing each version as a sequence of changes captured in the VGs component. By maintaining a history of graph summary versions, users can easily access and compare previous versions, as well as track changes and monitor the evolution of the data over time.

This algorithm initializes an empty list H to store the history of graph summary versions and an empty list V to store the sequence of graph summary versions. It then creates the initial version of the graph summary GS using the Operator O and the original Data Graph DG, adds the initial version to V, and creates a quadruplet V_Gs representing the current version of the graph summary. This quadruplet is added to H. The algorithm then makes changes to the original Data Graph DG, creates a new version of the graph summary GS using the Operator O and the updated Data Graph DG, and adds the new version to V. It also updates the current timestamp and creates a new quadruplet V_Gs representing the current version of the graph summary, which is added to H. Finally, the algorithm returns the final V_Gs with the history of graph summary versions.

Algorithm 7: Summary Versioning Process

Input : DG:original Data Graph
Output: $v_$GS: $GraphSummaryversionwithhistory$
Local Variables: $Operator(o), t : Currenttimestamp, H$

Initialize an empty list H to store the history of graph summary versions
Initialize an empty list V to store the sequence of graph summary versions
Create the initial version of the graph summary GS using the Operator O and the original Data Graph DG
Add the initial version to V
Create a quadruplet $V\_Gs = (DG, O, V, t)$ representing the current version of the graph summary
Add V_Gs to H
`// Repeat the following steps as needed`
Make changes to the original Data Graph DG.
Create a new version of the graph summary GS using the Operator O and the updated Data Graph DG
Add the new version to V
Update the current timestamp
Create a new quadruplet$V_Gs = (DG, O, V, t)$ representing the current version of the graph summary
Add V_Gs to H.
Return the final V_Gs with the history of graph summary versions.

9. Experimentation

9.1. Implemented Scenario description

To validate our proposed approach, we tested it on Type 2 Diabetes Monitoring system (T2DM system) Monitoring scenario presented in Section 2 to show how our approach intends to overcome scenario limits mentioned in this study. For our implementation, we developed an Angular-Python Framework called DGsum that implements the services described in this paper. The motivating scenario presented in section 2 was used as a prototype for our implementation, and we utilized a diverse dataset that focused on pregnant women with diabetes. The experimental protocol aims to tackle the difficulty of managing diverse and heterogeneous data from various sources, including medical devices, by providing GP with effective tools to interpret the data according to their specific requirements. To achieve this goal, the protocol entails integrating heterogenous data and medical devices to improve interoperability and enforce data standardization. Furthermore, it involves developing advanced data management and real time analysis tools that can handle the massive amount of ambiguous data. Our evaluation design was based on the implemented scenario in a Service Oriented Architecture (SOA) 3 as an evolution of the Component Based Architecture, Interface Based Design (Object Oriented), and Distributed Systems. SOA provides a simple and scalable paradigm for organizing large networks of systems that require interoperability to realize the value inherent in the individual components. By minimizing trust assumptions that are often implicitly made in smaller scale systems, SOA is scalable and manageable. As architects using SOA principles, we are better equipped to develop systems that are scalable and manageable.

9.2. Experimental Protocol

Experimental Objectives

– **Objective 1: Qualitative Evaluation** To evaluate the usability and user experience of the DGsum system and its impact on data analysis for healthcare professionals. This protocol involved GP with experience in data analysis and diabetes management, who

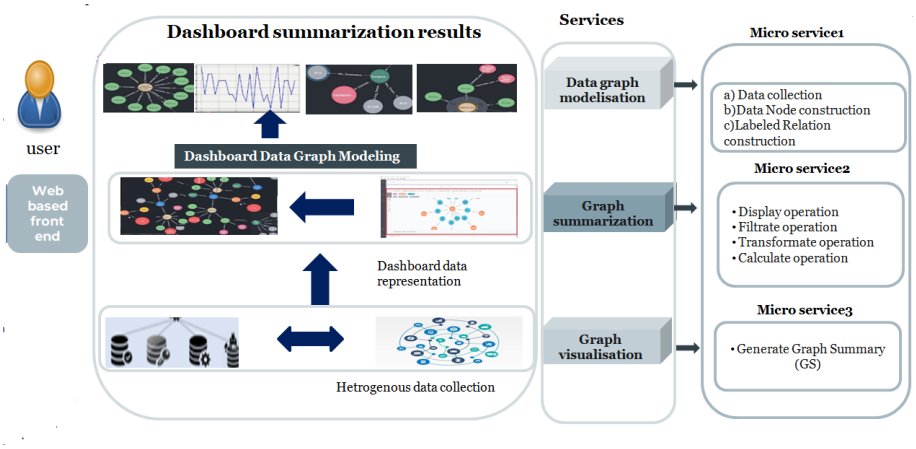

Fig. 3. Proposed technical Architecture oriented services of our system

was asked to perform specific tasks using the DGsum system, such as creating and visualizing a data graph or summarizing data based on a specific query. Participants was asked to provide feedback on the ease of use of the system, the usefulness of the tools provided, and their overall satisfaction with the system. The data collected from this protocol helped to identify any usability issues and inform future improvements to our system.

– **Objective 2: Quantitative Evaluation** To evaluate the effectiveness of the DGsum system in improving patient outcomes in pregnant women with diabetes, we evaluated the performance of our algorithms in terms of runtime, coverage, and loss of information of answers to queries on data graphs and graph summaries. We also conducted user studies to evaluate the effectiveness of the system in supporting the task of analyzing patient health records.

Dataset The dataset for pregnant women with diabetes contains various types of health data, including demographic information such as age, gender, and race, as well as medical history, surgical history, family history, and medication lists. The dataset also includes numerical data such as blood sugar levels, temperature, blood pressure, and BMI, which provide important information about the patient's diabetic condition. In addition to these simple numerical data, the dataset also contains complex data in the form of an ECG recording, Xray image, ultrasound image, MRI scan, blood test results, urine test results, and a symptoms diary. These data provide more detailed and specific information about the patient's health status and can be used to track their progress over time. Overall, the dataset provides a comprehensive view of the patient's health, which can be used by healthcare providers to inform their care and treatment plan. The combination of simple numerical data and complex data allows for a more complete understanding of the patient's health, enabling healthcare providers to make informed decisions about their care.

Participants The experiment are conducted with a group of healthcare professionals, including general practitioners and medical specialists. The participants should have basic knowledge and experience in interpreting medical data.

Procedure

- Participants were given a brief introduction to the DGsum system and the scenario presented in section 2 of the paper.
- Participants were provided access to the system and dataset, and they were asked to create a data graph that summarizes the health data of a pregnant woman with diabetes.
- Participants were asked to visualize the data graph and perform various operations, such as filtering, sorting, and grouping.
- Participants were asked to interpret the data and identify any potential health risks or concerns
- The participants were timed during the experiment, and their progress was recorded.
- After the experiment, participants were asked to provide feedback on the usability and effectiveness of the system, as well as any suggestions for improvement.

Data Analysis

- The time taken by the participants to create the data graph and perform the required operations was recorded and analyzed
- The accuracy and completeness of the data graph created by the participants was assessed.
- The feedback provided by the participants was analyzed and used to improve the system.

Expected outcomes

- The experimental protocol aimed to evaluate the effectiveness of the DGsum system in managing diverse and heterogeneous data from various sources
- The results of the experiment were analyzed to gain insights into the usability and effectiveness of the system and its potential to enhance patient care
- The feedback provided by the participants was analyzed and used to improve the system and make it more user-friendly and effective

Graphs generation

- **Data Graph Generation** To conduct an experimentation with the scenario presented in section 2, we have generated the DG of the pregnant women with diabetes. The generated DG includes:

  - **Data Nodes (DN):**
    - *Demographics Data:* represents demographic data about the patient, such as her age, gender, race, and other relevant information.

- *Medical History:* represents previous medical conditions, surgeries, and procedures they she undergone, as well as allergies, history may include details about the patient's lifestyle habits, such as smoking or alcohol consumption, which could impact her overall health.
- *Surgical History:* represents information about the surgical procedures that the patient has undergone in the past, including the date of the procedure, the name of the procedure, the surgeon who performed it, and any relevant details about the procedure, such as complications or outcomes. It also includes information about anesthesia and post-operative care.
- *Family History:* represents the patient's family medical history, including any relevant genetic or hereditary conditions.
- *Medication Lists:* represents a list of medications the patient is currently taking, including dosages and frequencies.
- *Blood Sugar Levels:* represents the patient's blood sugar levels, which are a key indicator of his diabetic condition.
- *Temperature:* represents the patient's body temperature
- *Blood Pressure:* represents simple numerical data about the patient's blood pressure.
- *BMI:* represents simple numerical data about the patient's body mass index.
- *Electrocardiogram (ECG):* represents complex data in the form of an ECG recording.
- *Xray:* represents complex data in the form of an Xray image.
- *Ultrasound:* represents complex data in the form of an ultrasound image.
- *Magnetic Resonance Imaging (MRI):* represents complex data in the form of an MRI scan.
- *Blood Test Results:* represents complex data in the form of blood test results, including various numerical and textual data points.
- *Urine Test Results:* represents complex data in the form of urine test results, including various numerical and textual data points.
- *Symptoms Diary:* represents complex data in the form of a diary of the patient's symptoms and his severity over time.

- **Relations (R):**
  Here are all relationships with corresponding labels between the different DN:
  - *"hasRecentBloodSugarLevel":* Demographics Data→Blood Sugar Levels
  - *"has RecentBodyTemperature":* Demographics Data→ Temperature
  - *"hasRecentBloodPressure":* Demographics Data→Blood Pressure
  - *"hasRecentBMI":* Demographics Data→hasRecentBMI
  - *"hasSurgicalHistory":* Medical History → Surgical Histories
  - *"has PastMedicationList":* Medical History →Medication Lists
  - *"hasPastbloodTestResult":* Medical History→ Blood Test Results
  - *"hasPastUrineTestResult":* Medical History → Urine Test Results
  - *"hasSymptomsDiary":* Medical History → Symptoms Diary
  - *"has ECG":* Medical History → Electrocardiogram
  - *"has Xray":* Medical History → Ultrasound
  - *"has MRI":* Medical History → Magnetic Resonance Imaging
  - *"hasSurgicalMedicationList":* Surgical Histories → Medication Lists

- *"has SurgicalBloodTestResult":* Surgical Histories →Blood Test Results
- *"haSurgicalUrineTestResult":* Surgical Histories → Urine Test Results
- *"hasMedicationBloodTestResult":* Medication Lists→Blood Test Results
- *"hasMedicationUrineTestResult":* Medication Lists→Urine Test Results
- *"hasBloodUrineTest": :* Blood Test Results → Urine Test Results

We generated in figure 4 the Data Graph (DG) for the pregnant woman with diabetes as described in the motivating scenario in section 2. The DG includes both simple and complex DN that provide information about her health condition. Simple DNs, such as demographic data, blood pressure, and body mass index, represent numerical or basic information. Meanwhile, complex DNs, such as ECG recordings, X_ray and ultrasound images, and MRI scans, contain more intricate data that require specialized interpretation. To depict the relationships between the different DNs, relations (R) were established for example, "hasRecentBloodSugarLevel" connects Demographics Data to Blood Sugar Levels DN. Other relationships connect simple DNs, such as blood sugar levels and temperature, to the Demographics DN. Other relationships link medical history to surgical history, medication lists, blood and urine test results, and symptoms diary. These connections between DNs help to provide a comprehensive view of the patient's health data and enable the identification of potential health risks or concerns.

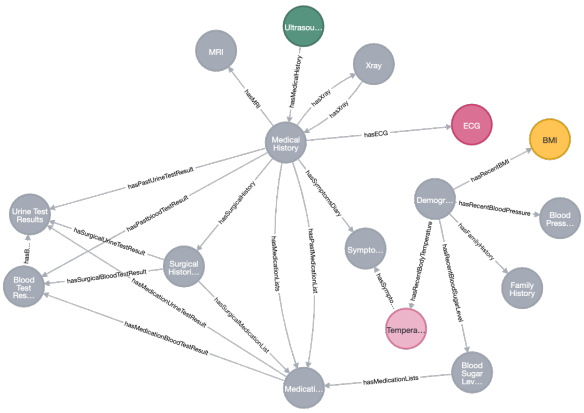

Fig. 4. The DG related to the pregnant women

Graph Summary Generation Table 4 presents the different types of queries that we will be using to extract and analyze the patient's medical data. These queries include transformation queries, which extract specific data points and present them in a certain way, calculation queries, which perform calculations on the extracted data, filtering queries, which narrow down the data based on certain criteria, and display queries, which present the data in an easy−to−understand format. By using these queries, we aim to generate comprehensive summaries of the patient's medical history and progress, which will inform their treatment plan.

Table 4. Querie types for graph summary generation

Operation	***Query***	***Description***
(1)Transformation	Query for blood sugar levels over time (Q1.1)	This query aims to extract the blood sugar levels and create a line chart to show how the levels have changed over time
	Query for blood and urine test results (Q1.2)	This query aims to extract the numerical and textual data points and create a table to show results for each test.
	Query for symptoms diary (Q1.3)	This query aims to extract the medication names and create a chart to show haw the symptoms have changed over time.
	Query for medication usage (Q1.4)	This query aims to extract the symptom description and severities and create a line chart to show how the symptoms have been prescribed.
	Query for surgical history (Q1.5)	This query aims to extract the surgical procedures and their dates and create a timeline to show when each procedure was performed
	Query for family history (Q1.6)	This query aims to extract the relevant data and create a table to show which conditions are present in the patient's family history
(2)Calculation	Calculate the correlation coefficient between blood sugar and BM (Q2.1)	This query aims to extract the blood sugar levels and BMI values and calculate the correlation coefficient between the two varaiables
	Calculate the percentage change in blood pressure (Q2.2)	This query aims to extract the blood pressure values from the Blood Pressure DN and calculate the percentage change from the previous reading.
	Calculate the average frequency of symptoms (Q2.3)	This query aims to extract the symptom data from the Symptoms Diary DN and calculate the average frequency of each symptom over a specified time period

(3) Filtering	Filter The MRI scans by body part and data range(Q3.1)	This query aims to extract the relevant MRI scan data and filter them to display only those for a specific body part and taken within a specified date range
	Filter the blood sugar levels by time of day (Q3.2)	This query aims to extract the blood sugar levels based on the time of day.
	Filter the medical records by data range (Q3.3)	This query aims to retrieve relevant medical information within a designated date interval.
	Filter the symptoms diary by severity (Q3.4)	This query aims to extract the relevant MRI scan data and filter them to display only those for a specific blood part and taken within a specified date rang
(4) Display	Display the Patient' s information (Q4.1)	This query would display the patient's demographics data, blood sugar levels, and symptoms diary
	Display the medical list (Q4.2)	This query aims to display all the medication lists.
	Display the blood test result (Q4.3)	This query aims to display the blood test result for the pregnant women with diabetes
	Display the ultrasound images (Q4.4)	This query aims to display the ultrasound images for the pregnant women with diabetes that were taken during the second trimester.
	Dispaly the surgical History (Q4.5)	This query aims to display the patient's surgical history, including the name of the surgery, the date it was performed, and the name of the surgeon.

We have provided some examples of queries that utilize the graph summary (GS) results to visualize the output. As shown in Figure 6, a doctor may request to visualize a patient's information, which falls under the category of display queries. On the other hand, Figure5 depicts a graph that displays various data nodes filtered based on their connection to surgical histories. This graph is likely generated as a result of a query requested by a GP and falls under the category of filtering queries. Figure 6 depicts a user requesting the summarization and visualization of numerical data nodes related to temperature measurements. The doctor would be able to interpret the curve represented by the variation in temperature. On the other hand, Figure 5 shows a query that involves extracting numerical data nodes related to temperature measures, and calculating the maximum, minimum, and average values of this measure. This query would enable the doctor to analyze and un-

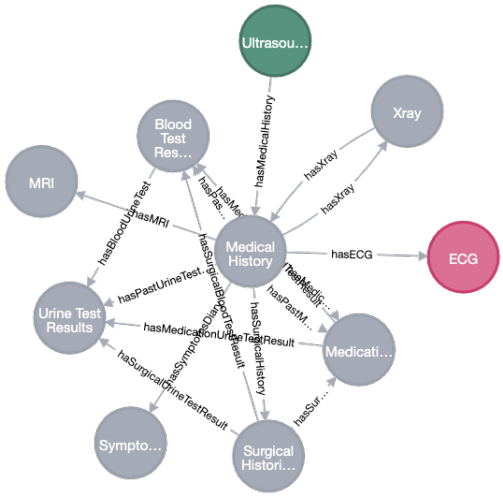

Fig. 5. Generating Summary Graph for query (Q3.3)

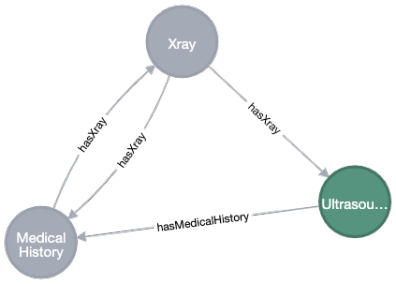

Fig. 6. Generating Summary Graph for query (Q2.3)

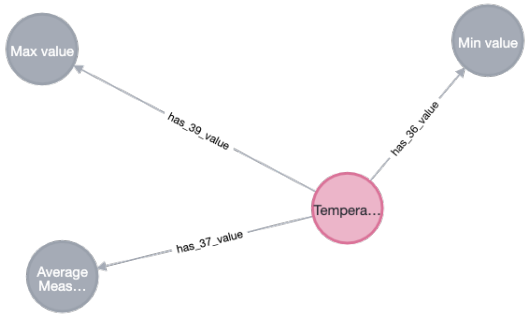

Fig. 7. Generating Summary Graph for query (Q1.3))

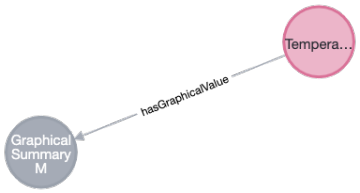

Fig. 8. Generating Summary Graph for query (Q4.3)

derstand the patient's health condition in relation to the given measure. These results fall under the category of calculation operation queries. 8 illustrate the query that synthesis a specific type of date like an image during a specific periode.

10. Evaluation

10.1. Qualitative Evaluation

Evaluation Scenario Ten participants were instructed to complete several tasks using the our system, including creating and customizing data graphs, summarizing data based on specific queries, and exporting data graphs in various formats. Following this, we conducted individual interviews with participants to obtain qualitative feedback on their experience using the system. Additionally, we administered a survey to gather feedback on participant satisfaction, ease of use, and suggestions for improvement. During the interviews, we utilized open-ended questions such as "How satisfied were you with your experience using the DGsumm system?" and "Did you find the data summarization feature useful? Why or why not?" to encourage participants to provide detailed feedback and express their thoughts and opinions. These were just a few examples of the questions asked, but the goal was to allow participants to share their experiences freely. The interview and survey data were analyzed to identify common themes and areas for improvement. Participants' behavior during the tasks was also observed, and any usability issues or roadblocks encountered were noted. The data analysis results were compiled into a comprehensive evaluation report, which included recommendations for improving the DGsmm system based on participant feedback and observations. Finally, based on the evaluation report, recommended improvements were implemented to enhance the DGsum system's usability and user experience.

Evaluation Report After analyzing the data, we identified the following common themes:

- *Overall Satisfaction:* All participants reported a high level of satisfaction with the system. They appreciated the ease of use of the tools provided and found the system to be intuitive and user-friendly.
- *Ease of Use:* Participants praised the system's easy-to-use interface, which allowed them to create and visualize data graphs quickly. They also found the summarization tool to be helpful in summarizing large data sets.
- *Suggestions for Improvement:* Participants suggested several areas for improvement, including the need for more customization options for visualizations, the ability to

export data graphs in various formats, and the addition of more advanced analysis tools.

Based on this feedback, we identified several areas for improvement, such as expanding customization options and adding more advanced analysis tools. We also plan to implement the ability to export data graphs in various formats to address a common suggestion for improvement.

10.2. Quantitative Evaluation

Evaluation metrics and results In the field of graph summarization, the evaluation metrics remain a challenge, and multiple metrics can be employed to assess the quality of graph summarization, depending on the specific objectives of each task. Evaluating the quality of a graph summary may require using multiple metrics to evaluate both the structure and content.In our work we used three metrics, include:

- **Calculation time**: Measures the speed at which the graph summary method can produce a result for a given input data graph.
- **Loss of information**: Measures the number of nodes and relation in original graph that are preserved in the summary result.

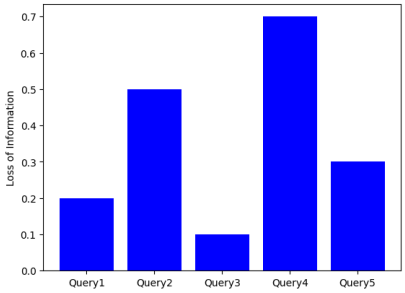

Fig. 9. Loss information behavior during Display operation

Discussion The objective of the initial evaluation was to confirm the extent of information loss that occurred during the summarization process. The tests indicated that the degree of information loss in the graph summary was minimal when compared to the original data graph and was dependent on the query objective

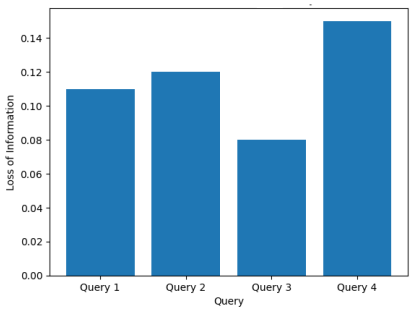

Fig. 10. Loss information behavior during filtering operation

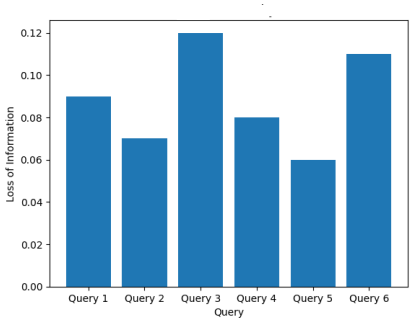

Fig. 11. Loss information behavior duringTransformation operation

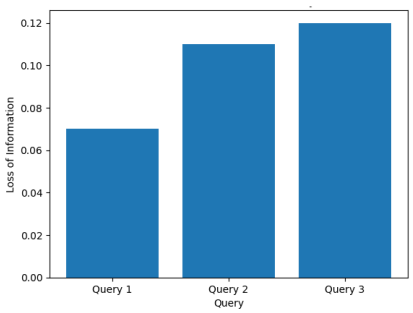

Fig. 12. Loss information behavior during Calculation operation

Table 5. Table of query input and output

Operation	Query	DN source	Output
(1) Transformation	Q1.1	Blood sugar level	Image (line chart)
	Q1.2	Blood test result and Urine result	Table
	Q1.3	Symptoms Diary	Image
	Q1.4	Medication list	Image
	Q1.5	Surgical history	Image (timeline)
	Q1.6	Family history	Table
(2) Calculation	Q2.1	Blood sugar Level and BMI	Numeric (correlation value)
	Q2.2	Blood presure	Numeric (percentage value)
	Q2.3	Symptom Diary	Numeric (average value)
(3) Filtering	Q3.1	Blood sugar level	Numeric
	Q3.2	Medical Data	Text
	Q3.3	Symptom Diary	Text
	Q3.4	MRI	Image
(4) Dispaly	Q4.1	EHR	Text
	Q4.2	EHR	Numeric
	Q4.3	EHR	Image
	Q4.4	EHR	Text
	Q4.5	EHR	Graphical

The information loss resulting from the transformation operation is shown in Figure 11. The results indicate that Query (Q1.3) experiences the highest degree of information loss with 0.12, followed by Query 6 with a loss of 0.11 percent. Query(Q1.1) and Query Q1.4) both exhibit moderate information loss with 0.085 and 0.07, respectively. The query with the lowest information loss is Query(Q1.2), with a value of 0.063. Regarding the second category Display operation shown in Figure 9, it is worth noting that query (Q4.4) had the highest information loss. This is because the objective of this query was to analyze three concepts: demographics data, blood sugar levels, and symptoms diary, and to display only the relevant information in the summary. Moving on to the third category Filtering operation shown in Figure 10, we observed that queries(Q3.2) and (Q3.4) had the highest information loss. This can be attributed to the fact that the filtering operation based on a specified date range may not capture important context or details that are relevant to the medical records, and therefore more data might be lost.

Finally, in the last category, i.e., Calculating operation shown in Figure 12, we found that queries(Q2.2) and (Q2.3) had more information loss than query(Q2.1). While synthesizing the percentage or the average may be relevant for visualizing and interpreting the results, there is no guarantee that information loss will not occur.

In the second metric, we analyzed the run time of various queries for different types of operations performed on the graph summary. We categorized the operations into four categories: filtering, calculation, transformation, and display.
In the filtering category, as shown in Figure 13, query (Q3.4) took the most time (28ms) to generate the result, while queries (Q3.1) and (Q3.3) took approximately the same amount of time. This order is because filtering by body part and date range and filtering med-

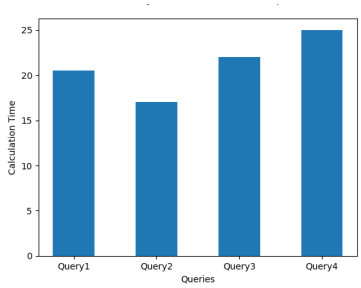

Fig. 13. Run Time behavior during Filtering operation

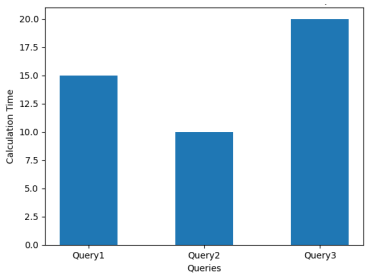

Fig. 14. Run Time behavior during Calculation operation

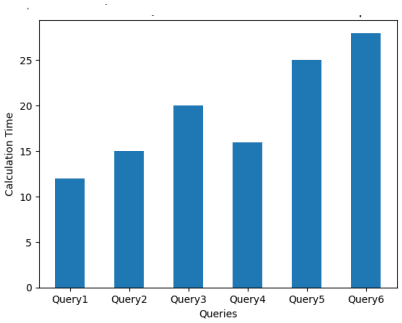

Fig. 15. Run Time behavior during Transformation operation)

ical records by date range involve a large amount of data and require complex queries involving multiple data sources.

In the calculation category, as shown in Figure 14, query(Q2.3) took the most time (21ms) because calculating the correlation coefficient involves a more complex statistical calculation than the other two queries. Calculating the average frequency of symptoms also involves some complexity, as it requires aggregating and analyzing a large amount of symptom data, which took 15ms. Query (Q2.2) is the least complex of the three queries, taking only 10ms.

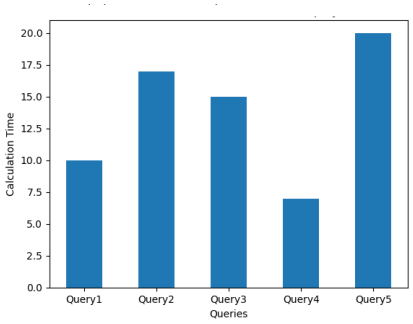

Fig. 16. Run Time behavior during Display operation

In the display category, as shown in Figure 16, we analyzed five queries. Query (Q4.1) took the most time, as it involves extracting and joining data from multiple DNs, such as demographics data, blood sugar levels, and symptoms diary, which could take more time to execute. Query(Q4.2) involves retrieving and displaying large files of ultrasound images, which could take more time to load and display compared to other types of data. Query (Q4.3) focuses on filtering based on multiple criteria, which could also take significant time to execute. Query (Q4.4) involves simply displaying data from the Medication Lists DN, which is likely to be relatively quick to execute compared to the other queries.

In the transformation category, as shown in Figure 15, we found that query (Q1.6) took the most time (28ms). The result of this query is creating a table that shows which conditions are present in the patient. This would likely be the most time-consuming query, as it involves sorting through potentially large amounts of data and compiling it into a table. Query (Q1.1) aims to extract blood sugar levels from the Blood Sugar Levels DN and create a line chart that shows how the levels have changed over time. While this query may still take some time to summarize due to the need to process and plot the data, it may be quicker than the other queries due to the relatively narrow focus of the data extraction.

Overall, the run time of the queries varied depending on the type of operation performed and the complexity of the query.

11. Conclusion

In this paper, we extensively studied utility-driven data graph modeling and graph summarization and made several innovative contributions. Using user queries, we introduced two new lossless graph summaries: a structured one and a content-based one. Furthermore, we illustrated our approach by integrating the proposed data graph formalism into heterogeneous input data. We proposed several primary operations for the summarization process and conducted experiments to design a lossy summarization algorithm based on two metrics: running time and information loss. The aim was to validate our proposed scenario for the medical domain.

The perspectives of this work are promising. The DGsumm system has shown to be effective in summarizing heterogeneous data graphs while maintaining a low degree of information loss. The qualitative evaluation feedback has provided valuable insights into

the user experience and suggestions for improvements that could be implemented in future iterations of the system. The quantitative evaluation metrics have demonstrated the system's capacity to handle user queries and generate graph summaries in a timely manner.

The DGsumm system has the potential to be applied in various domains, including finance, healthcare, and social media, where large and complex data graphs are prevalent. Future work could involve expanding the system's capabilities to handle more complex queries and provide more advanced analysis tools. Additionally, integrating machine learning algorithms could improve the system's ability to personalize summarization based on individual user preferences. Overall, the DGsumm system has shown to be a promising approach for personalized summarization of heterogeneous data graphs with potential for further development and applications.

References

1. Abu-Faraj, Z.O., Barakat, S.S., Chaleby, M.H., Zaklit, J.D.: A sim card-based ubiquitous medical record bracelet/pendant system—a pilot study. In: 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI). vol. 4, pp. 1914–1918. IEEE (2011)
2. Adhikari, B., Zhang, Y., Amiri, S.E., Bharadwaj, A., Prakash, B.A.: Propagation-based temporal network summarization. IEEE Transactions on Knowledge and Data Engineering 30(4), 729–742 (2017)
3. Aggarwal, C.C., Wang, H.: A survey of clustering algorithms for graph data. Managing and mining graph data pp. 275–301 (2010)
4. Ahmed, M.: Data summarization: a survey. Knowledge and Information Systems 58(2), 249–273 (2019)
5. Ahmed, M., Mahmood, A.N., Islam, M.R.: A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems 55, 278–288 (2016)
6. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data mining and knowledge discovery 29, 626–688 (2015)
7. Angles, R., Gutierrez, C.: Survey of graph database models. ACM Computing Surveys (CSUR) 40(1), 1–39 (2008)
8. Bade, R., Schlechtweg, S., Miksch, S.: Connecting time-oriented data and information to a coherent interactive visualization. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 105–112 (2004)
9. Bashyam, V., Hsu, W., Watt, E., Bui, A.A., Kangarloo, H., Taira, R.K.: Problem-centric organization and visualization of patient imaging and clinical data. Radiographics 29(2), 331–343 (2009)
10. Bates, D.W., Ebell, M., Gotlieb, E., Zapp, J., Mullins, H.: A proposal for electronic medical records in us primary care. Journal of the American Medical Informatics Association 10(1), 1–10 (2003)
11. Boran, F.E., Akay, D., Yager, R.R.: An overview of methods for linguistic summarization with fuzzy sets. Expert Systems with Applications 61, 356–377 (2016)
12. Borland, D., West, V.L., Hammond, W.E.: Multivariate visualization of system-wide national health service data using radial coordinates. In: Proc. Workshop on Visual Analytics in Healthcare (2014)
13. Bui, A.A., Aberle, D.R., Kangarloo, H.: Timeline: visualizing integrated patient records. IEEE Transactions on Information Technology in Biomedicine 11(4), 462–473 (2007)
14. Cao, F., Estert, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proceedings of the 2006 SIAM international conference on data mining. pp. 328–339. SIAM (2006)

15. Carenini, A., Cerri, D., Krummenacher, R., Simperl, E.: Enabling interoperability of patient summaries across europe with triplespaces. In: Interoperability in Healthcare Information Systems: Standards, Management, and Technology, pp. 232–249. IGI Global (2013)
16. Čebirić, Š., Goasdoué, F., Kondylakis, H., Kotzinos, D., Manolescu, I., Troullinou, G., Zneika, M.: Summarizing semantic graphs: a survey. The VLDB journal 28, 295–327 (2019)
17. Chiarandini, L.: Human-centered exploration and discovery of content in large information spaces (2011)
18. Clayton, P.D., Narus, S.P., Bowes III, W.A., Madsen, T.S., Wilcox, A.B., Orsmond, G., Rocha, B., Thornton, S.N., Jones, S., Jacobsen, C.A., et al.: Physician use of electronic medical records: issues and successes with direct data entry and physician productivity. In: AMIA annual symposium proceedings. vol. 2005, p. 141. American Medical Informatics Association (2005)
19. Cook, D.J., Holder, L.B.: Graph-based data mining. IEEE Intelligent Systems and Their Applications 15(2), 32–41 (2000)
20. Crawford, P., Brown, B., Baker, C., Tischler, V., Abrams, B., Crawford, P., Brown, B., Baker, C., Tischler, V., Abrams, B.: Health humanities. Springer (2015)
21. Delong, A., Boykov, Y.: A scalable graph-cut algorithm for nd grids. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
22. Dunne, C., Shneiderman, B.: Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3247–3256 (2013)
23. Fan, W., McCloskey, J., Yu, P.S.: A general framework for accurate and fast regression by data summarization in random decision trees. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 136–146 (2006)
24. Fan, W., Li, J., Wang, X., Wu, Y.: Query preserving graph compression. In: Proceedings of the 2012 ACM SIGMOD international conference on management of data. pp. 157–168 (2012)
25. Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., Zhang, J.: Graph distances in the data-stream model. SIAM Journal on Computing 38(5), 1709–1727 (2009)
26. Gong, J.J., Guttag, J.V.: Learning to summarize electronic health records using cross-modality correspondences. In: Machine learning for healthcare conference. pp. 551–570. PMLR (2018)
27. Gunter, T.D., Terry, N.P.: The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. Journal of medical Internet research 7(1), e383 (2005)
28. Han, W., Miao, Y., Li, K., Wu, M., Yang, F., Zhou, L., Prabhakaran, V., Chen, W., Chen, E.: Chronos: a graph engine for temporal graph analysis. In: Proceedings of the Ninth European Conference on Computer Systems. pp. 1–14 (2014)
29. Harrington, J.L.: Relational database design and implementation. Morgan Kaufmann (2016)
30. Hirsch, J.S., Tanenbaum, J.S., Lipsky Gorman, S., Liu, C., Schmitz, E., Hashorva, D., Ervits, A., Vawdrey, D., Sturm, M., Elhadad, N.: Harvest, a longitudinal patient record summarizer. Journal of the American Medical Informatics Association 22(2), 263–274 (2015)
31. Hu, P., Lau, W.C.: A survey and taxonomy of graph sampling. arXiv preprint arXiv:1308.5865 (2013)
32. Huang, J., Abadi, D.J., Ren, K.: Scalable sparql querying of large rdf graphs. Proceedings of the VLDB Endowment 4(11), 1123–1134 (2011)
33. Hunter, J., Freer, Y., Gatt, A., Logie, R., McIntosh, N., Van Der Meulen, M., Portet, F., Reiter, E., Sripada, S., Sykes, C.: Summarising complex icu data in natural language. In: Amia annual symposium proceedings. vol. 2008, p. 323. American Medical Informatics Association (2008)
34. Jagadish, H., Ng, R.T., Ooi, B.C., Tung, A.K.: Itcompress: An iterative semantic compression algorithm. In: Proceedings. 20th International Conference on Data Engineering. pp. 646–657. IEEE (2004)
35. Kang, U., Faloutsos, C.: Beyond'caveman communities': Hubs and spokes for graph compression and mining. In: 2011 IEEE 11th international conference on data mining. pp. 300–309. IEEE (2011)

36. Kriegel, H.P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley interdisciplinary reviews: data mining and knowledge discovery 1(3), 231–240 (2011)
37. Lan, J., Song, Z., Miao, X., Li, H., Li, Y., Dong, L., Yang, J., An, X., Zhang, Y., Yang, L., et al.: Skin damage among health care workers managing coronavirus disease-2019. Journal of the American Academy of Dermatology 82(5), 1215–1216 (2020)
38. Lebanoff, L., Song, K., Liu, F.: Adapting the neural encoder-decoder framework from single to multi-document summarization. arXiv preprint arXiv:1808.06218 (2018)
39. Liang, J., Tsou, C.H., Poddar, A.: A novel system for extractive clinical note summarization using ehr data. In: Proceedings of the 2nd clinical natural language processing workshop. pp. 46–54 (2019)
40. Liu, H., Friedman, C.: Cliniviewer: a tool for viewing electronic medical records based on natural language processing and xml. In: MEDINFO 2004. pp. 639–643. IOS Press (2004)
41. Liu, J., Cao, Y., Lin, C.Y., Huang, Y., Zhou, M.: Low-quality product review detection in opinion summarization. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). pp. 334–342 (2007)
42. Lukas, P.S., Krummenacher, R., Biasiutti, F.D., Begré, S., Znoj, H., von Känel, R.: Association of fatigue and psychological distress with quality of life in patients with a previous venous thromboembolic event. Thrombosis and haemostasis 102(12), 1219–1226 (2009)
43. Maccioni, A., Abadi, D.J.: Scalable pattern matching over compressed graphs via dedensification. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1755–1764 (2016)
44. Matheny, M.E., Miller, R.A., Ikizler, T.A., Waitman, L.R., Denny, J.C., Schildcrout, J.S., Dittus, R.S., Peterson, J.F.: Development of inpatient risk stratification models of acute kidney injury for use in electronic health records. Medical Decision Making 30(6), 639–650 (2010)
45. McInerney, D.J., Dabiri, B., Touret, A.S., Young, G., Meent, J.W., Wallace, B.C.: Query-focused ehr summarization to aid imaging diagnosis. In: Machine Learning for Healthcare Conference. pp. 632–659. PMLR (2020)
46. Miotto, R., Li, L., Dudley, J.T.: Deep learning to predict patient future diseases from the electronic health records. In: European conference on information retrieval. pp. 768–774. Springer (2016)
47. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., Group*, P.: Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. Annals of internal medicine 151(4), 264–269 (2009)
48. Moradi, M., Ghadiri, N.: Different approaches for identifying important concepts in probabilistic biomedical text summarization. Artificial intelligence in medicine 84, 101–116 (2018)
49. Nallaperuma, D., De Silva, D., et al.: A participatory model for multi-document health information summarisation. Australasian Journal of Information Systems 21 (2017)
50. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Mining text data, pp. 43–76. Springer (2012)
51. Nielsen, F., Nielsen, F.: Hierarchical clustering. Introduction to HPC with MPI for Data Science pp. 195–211 (2016)
52. Pham, D.T., Dimov, S.S., Nguyen, C.D.: Selection of k in k-means clustering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 219(1), 103–119 (2005)
53. Pivovarov, R., Perotte, A.J., Grave, E., Angiolillo, J., Wiggins, C.H., Elhadad, N.: Learning probabilistic phenotypes from heterogeneous ehr data. Journal of biomedical informatics 58, 156–165 (2015)
54. Pouzols, F.M., Lopez, D.R., Barros, A.B.: Mining and Control of Network Traffic by Computational Intelligence, vol. 342. Springer (2011)
55. Powsner, S.M., Tufte, E.R.: Summarizing clinical psychiatric data. Psychiatric Services 48(11), 1458–1460 (1997)

56. Qu, Q., Liu, S., Jensen, C.S., Zhu, F., Faloutsos, C.: Interestingness-driven diffusion process summarization in dynamic networks. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14. pp. 597–613. Springer (2014)
57. Rabbouch, H., Saâdaoui, F., Mraihi, R.: Unsupervised video summarization using cluster analysis for automatic vehicles counting and recognizing. Neurocomputing 260, 157–173 (2017)
58. Radev, D., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Computational linguistics 28(4), 399–408 (2002)
59. Raghavan, S., Garcia-Molina, H.: Representing web graphs. In: Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405). pp. 405–416. IEEE (2003)
60. Reeve, L.H., Han, H., Brooks, A.D.: The use of domain-specific concepts in biomedical text summarization. Information Processing & Management 43(6), 1765–1776 (2007)
61. Rogers, J.L., Haring, O.M., Watson, R.A.: Automating the medical record: emerging issues. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. p. 255. American Medical Informatics Association (1979)
62. Shah, N., Koutra, D., Zou, T., Gallagher, B., Faloutsos, C.: Timecrunch: Interpretable dynamic graph summarization. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1055–1064 (2015)
63. Shahar, Y., Goren-Bar, D., Boaz, D., Tahan, G.: Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. Artificial intelligence in medicine 38(2), 115–135 (2006)
64. Shen, Z., Ma, K.L., Eliassi-Rad, T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. IEEE transactions on visualization and computer graphics 12(6), 1427–1439 (2006)
65. Shi, L., Tong, H., Tang, J., Lin, C.: Vegas: Visual influence graph summarization on citation networks. IEEE Transactions on Knowledge and Data Engineering 27(12), 3417–3431 (2015)
66. Soysal, E., Warner, J.L., Denny, J.C., Xu, H.: Identifying metastases-related information from pathology reports of lung cancer patients. AMIA Summits on Translational Science Proceedings 2017, 268 (2017)
67. Tang, N., Chen, Q., Mitra, P.: Graph stream summarization: From big bang to big crunch. In: Proceedings of the 2016 International Conference on Management of Data. pp. 1481–1496 (2016)
68. Tapak, L., Mahjub, H., Hamidi, O., Poorolajal, J.: Real-data comparison of data mining methods in prediction of diabetes in iran. Healthcare informatics research 19(3), 177–185 (2013)
69. Tian, Y., Patel, J.M.: Tale: A tool for approximate large graph matching. In: 2008 IEEE 24th International Conference on Data Engineering. pp. 963–972. IEEE (2008)
70. Toivonen, H., Zhou, F., Hartikainen, A., Hinkka, A.: Compression of weighted graphs. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 965–973 (2011)
71. Traverso, A., Van Soest, J., Wee, L., Dekker, A.: The radiation oncology ontology (roo): Publishing linked data in radiation oncology using semantic web and ontology techniques. Medical physics 45(10), e854–e862 (2018)
72. Vandenbroucke, J.P., Von Elm, E., Altman, D.G., Gøtzsche, P.C., Mulrow, C.D., Pocock, S.J., Poole, C., Schlesselman, J.J., Egger, M., Initiative, S.: Strengthening the reporting of observational studies in epidemiology (strobe): explanation and elaboration. PLoS medicine 4(10), e297 (2007)
73. Wang, Q., Laramee, R.S., Lacey, A., Pickrell, W.O.: Lettervis: a letter-space view of clinic letters. The Visual Computer 37(9), 2643–2656 (2021)
74. Wang, T.D., Plaisant, C., Shneiderman, B., Spring, N., Roseman, D., Marchand, G., Mukherjee, V., Smith, M.: Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. IEEE transactions on visualization and computer graphics 15(6), 1049–1056 (2009)

75. Wright, A., Pang, J., Feblowitz, J.C., Maloney, F.L., Wilcox, A.R., Ramelson, H.Z., Schneider, L.I., Bates, D.W.: A method and knowledge base for automated inference of patient problems from structured data in an electronic medical record. Journal of the American Medical Informatics Association 18(6), 859–867 (2011)
76. Xu, W., Lu, Z., Wu, W., Chen, Z.: A novel approach to online social influence maximization. Social Network Analysis and Mining 4, 1–13 (2014)
77. Yager, R.R., Ford, K.M., Cañas, A.J.: An approach to the linguistic summarization of data. In: Uncertainty in Knowledge Bases: 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'90 Paris, France, July 2–6, 1990 Proceedings 3. pp. 456–468. Springer (1991)
78. Zhang, N., Tian, Y., Patel, J.M.: Discovery-driven graph summarization. In: 2010 IEEE 26th international conference on data engineering (ICDE 2010). pp. 880–891. IEEE (2010)
79. Zhang, Z., Balay, J., Bertoldi, K., MaCoy, P.: Assessment of water capacity and availability from unregulated stream flows based on ecological limits of hydrologic alteration (eloha) environmental flow standards. River Research and Applications 32(7), 1469–1480 (2016)

Beldi Amal is a dedicated Ph.D. student, currently pursuing her doctoral studies through a cotutelle program between the University of El Manar in Tunisia and the University of Pau et des Pays de l'Adour. Her research focus lies in the realm of data graph modeling and summarization,visualization, topics modeling, and knowledge extraction. As a passionate scholar, she is actively involved as a member of the OpenCEMS chair. Alongside her academic pursuits, Amal also serves as an ATER (Attaché Temporaire d'Enseignement et de Recherche) at the University of Pau, further enriching her academic journey and contributing to the academic community.

Salma Sassi is an assistant professor of computer science with University of Jendouba, and member of DoCSYS research team at VNPC research and LIUPPA laboratory. She received her PhD in computer science from INSA Of Lyon, France and Manouba University, Tunisia in 2009. Her research interests include Representation and visualization of knowledges, indexing and semantic annotation, Healthcare Information Systems and Big Data.

Richard Chbeir is a professor of computer science at the University of Pau and the Adour Region in France, where he leads the computer science laboratory called LIUPPA. He is the director of the Semantics Privacy in Digital Ecosystems Research group (SPiDER).He is currently working on information and knowledge extraction. Chbeir is the head of the OpenCEMS industrial chair, He received his Ph.D. in computer science from the Institut national des sciences appliquées de Lyon in 2001 and got his Habilitation degree in 2010 from the University of Burgundy.

Abderrazak JEMAI Professor at National Institute of Applied Science and Technology (INSAT), University of Carthage, Tunis, Tunisia and former General Director of CNI (National Center of Informatics in Tunisia) dealing with government' applications and related access services, data storage and security (eGov, Intranet, etc.).His last interest is focused on the design (codesign of smart embedded systems) and the deployment of Smart-IoT (Smart Internet of Things) devices in the context of Smart City, Cloud Computing, Smart-Industry (Industry 4.0) and Big Data.

Received: March 31, 2023; Accepted: June 01, 2023.