A heterogeneous multi-modal medical data fusion framework supporting hybrid data exploration

Yong Zhang¹, Ming Sheng¹*, Xingyue Liu², Ruoyu Wang², Weihang Lin³, Peng Ren¹, Xia Wang⁴, Enlai Zhao⁵ and Wenchao Song⁶

Abstract
Industry 4.0 era has witnessed that more and more high-tech and precise devices are applied into medical field to provide better services. Besides EMRs, medical data include a large amount of unstructured data such as X-rays, MRI scans, CT scans and PET scans, which is still continually increasing. These massive, heterogeneous multi-modal data bring the big challenge to finding valuable data sets for healthcare researchers and other users. The traditional data warehouses are able to integrate the data and support interactive data exploration through ETL process. However, they have high cost and are not real-time. Furthermore, they lack of the ability to deal with multi-modal data in two phases—data fusion and data exploration. In the data fusion phase, it is difficult to unify the multi-modal data under one data model. In the data exploration phase, it is challenging to explore the multi-modal data at the same time, which impedes the process of extracting the diverse information underlying multi-modal data. Therefore, in order to solve these problems, we propose a highly efficient data fusion framework supporting data exploration for heterogeneous multi-modal medical data based on data lake. This framework provides a novel and efficient method to fuse the fragmented multi-modal medical data and store their metadata in the data lake. It offers a user-friendly interface supporting hybrid graph queries to explore multi-modal data. Indexes are created to accelerate the hybrid data exploration. One prototype has been implemented and tested in a hospital, which demonstrates the effectiveness of our framework.

Keywords: Heterogeneous data, Multi-modal medical data, Data fusion, Hybrid data exploration

Introduction
Nowadays, enormous amounts of multi-modal and heterogeneous medical data are generated every day from various medical devices and healthcare events, and fragmented in different organizations. These medical data include structured data such as Electronic Medical Records (EMRs), semi-structured data such as Comma-Separated Values (CSV), Extensible Markup Language (XML) and JavaScript Object Notation (JSON), and unstructured data [1] such as magnetic resonance imaging (MRI) scans, computerized tomography (CT) scans, X-ray and positron emission tomography (PET) scans, etc. We call the dataset including all of the three kinds as multi-modal data. The unstructured imaging data generated in one modern hospital could reach hundreds of terabytes. The volume and complexity of medical multi-modal data are increasing rather rapidly. The unstructured medical data usually have high dimensions, which makes it difficult to interpret and further organize and analyze to obtain more valuable information. Similarity search is a very important operation for data fusion, analytic technologies and knowledge graph construction of the unstructured medical data [2].

With the variety and diversity of medical data, data-driven decision-making approaches have been conducted for many years. In traditional computer-aided medical expert systems, decision-making is assisted by employing feature-level fusion or rule-based reasoning [3–5]. These computer-aided diagnosis systems focus on
semantic perception and entity association mining with the medical big data. Artificial intelligence techniques are also implemented in recent studies [6–8], including deep neural networks for feature extraction and image classification [6, 7, 9], and ensemble learning for classification of imbalanced streaming data [10]. There are also frameworks proposed based on Machine Learning (ML) or Deep Learning (DL) to extract the complementary information for multi-modal data [7, 11–13]. To deal the data fragmentation, panoramic interactive decision-making was proposed [3].

Data exploration is critical for medical researches. There are various platforms to integrate data and then perform data exploration [14–16]. In terms of medical information statistics, The Tuberculosis Data Exploration Portal (TB DEPOT) can be used to query TB patient cases, create and save patient queues, and perform comparative statistical analysis as required [14]. To support clinical decision-making, mathematical model prediction is integrated into the routine workflow on hematology oncology [15]. For medical analysis, Heart Failure Integrated Platform (HFIP) is built for data exploration, fusion analysis and visualization through the collection and strategic presentation of multi-modal data and several public knowledge bases [16].

However, these methods did not consider the fusion of heterogeneous multi-modal data by using a unified format and how to efficiently store them. Even with panoramic decision-making, state-of-the-art techniques do not provide user-friendly hybrid data exploration based on the fused data. Therefore, the capability of data analytics is limited without the supporting of hybrid data explorations.

Contributions In this paper, we propose a framework based on data lake supporting multi-modal medical data fusion and hybrid data exploration. The contributions are summarized as follows:

1. We implement the fusion of multi-modal data based on data lake. The multi-modal data, including structured, semi-structured and unstructured data, are transformed into one unified format for persistence.
2. We use a dynamic approach to manage multi-modal data according to various demands in medical field. The majority of the data is not shifted to the data lake. Based on the metadata, we can extract the data dynamically to fuel various applications.
3. We propose an index construction strategy on multi-modal data. These indexes are in accordance with the characteristic of multi-modal data.
4. We implement a method to manage the multi-modal indexes. Multi-level indexes are used according to the requirements of hybrid data exploration.
5. We propose a data exploration mechanism based on hybrid query and graph query for multi-modal data. One prototype has been implemented and tested in on hospital.

In the rest of this paper, we present the details as follows. Section 2 introduces related work. Section 3 presents the architecture of our framework. Section 4 describes the design of heterogeneous multi-modal medical data fusion and management. Section 5 discusses the dynamic index construction method and management strategy on multi-modal medical data. Section 6 describes how to do hybrid data exploration by demonstrating several medical scenarios. Section 7 shows the prototype implementation of our design and the application of the prototype. Section 8 summarizes the paper.

Related work
In this section, we discuss the researches on unstructured data storage, data exploration, embedding, indexing and data lake.

For unstructured multi-modal data, researchers use machine learning tools to extract the features [6–8]. Jiang et al. developed an advanced model to measure the association between latent topics [17]. There are also researches about unstructured data storage systems.

Peihao et al. [18] proposed a system to store medical unstructured data and generate the structured form for the various examination items in a report according to the constructed medical dictionary. However, this system focuses on storing information extracted from unstructured data instead of storing unstructured data. Yingcheng et al. [19] proposed INSMA, a system transforming medical data from different patient monitoring devices into a big text file including the monitoring results. An intelligent system is proposed for patients’ similar text information retrieval by information from template, topic, and latent semantic indexes [20]. However, none of these researches focuses on the fusion of multi-modal data. Their storage strategies are very limited, and their unstructured data do not include medical images.

Traditionally, a data exploration method concerns textual information retrieval [21], using both statistical approaches based on term frequency and semantic approaches with embedding information from WordNet and domain ontology. A similar framework is also introduced in [22], utilizing original lexical resources and word embedding information concerning question answering problem. For heart failure, HFIP is designed for data exploration and data analysis after the harvest of multi-modal data and several public knowledge bases [16]. However, these data exploration methods were not
based on the data fusion of multi-modal data, which restricted their capabilities to deal with the complicated medical scenarios.

Embeddings on different models represent different aspects of information of multi-modal data. Jiahua et al. [23] proposed a deep architecture enhanced with character embeddings and neural attention to improve the performance of hay fever-related content classification from Twitter data and the study is a step forward towards improved real-time pollen allergy surveillance from social media with state-of-art technology. Wei et al. [24] proposed a novel graph embedding framework, Adversarial and Random Walk Regularized Graph Embedding (ARWR-GE), and the results demonstrate that the framework achieves better performance than state-of-the-art graph embedding algorithms. Similarly, different index construction methods support different datasets. But there is a lack of researches to manage these embedding models and index construction methods in application scenarios.

The concept of Data Lake was first put forward to deal with the challenges brought by Data Warehouse. Data Lake technology has been employed in medical field. For example, Alhroob et al. [25] designed a data management framework based on Data Lake for semi-structured data of cardiovascular and cerebrovascular diseases by using k-means clustering with categorical and numerical data with big data characteristics. Kachaoui et al. [26] came up with a Data Lake framework combining semantic web services (SWS) and big data features in order to predict the case of coronavirus disease 2019 (COVID-19) in real-time. However, these data management frameworks in medical field based on Data Lake technology lack of extensibility and processing efficiency.

We use a table to compare our framework with other similar methods (See Table 1). There are five dimensions in the table: feature extraction (FE), multi-modal data fusion (MDF), multi-modal data management (MDM), multi-modal data index (MDI), and hybrid data exploration (HDE), in which a plus sign indicates that this function has been implemented in the paper mentioned in the first column.

In summary, we address the challenges as follows:

1. There are many medical data sources with rich and diverse structures, but lacking of feasible method for data fusion hinders further data analysis. How can we fuse the multi-modal medical datasets?
2. The requirements for medical datasets are constantly changing because of different research and application purposes. How can we get the multi-modal datasets flexibly?

3. The amount of multi-modal medical data is accumulating rapidly. How can we quickly search the datasets according to the characteristics of various data?
4. The order of the indexes used in data exploration influence the efficiency greatly. How can we manage indexes in a well-organized and efficient way according to the needs of medical data exploration?
5. Traditional data exploration methods lack of the capability to deal with multi-modal data. How can we use the fused results to support hybrid data exploration?

In the following sections, we will show how to cope with these challenges based on data lake.

### Architecture design

Our framework is established on data lake that is based on Apache Spark. The framework utilizes data lake for multi-modal data fusion, and develops dynamic data storage management and index management strategies, and supports hybrid data exploration. The architecture of our framework is presented in Fig. 1. It mainly consists of three components: a data lake layer for the dynamic, reliable and large scale storage, a dynamic index management layer for efficient management of multi-modal data, and a hybrid data exploration layer for heterogeneous multi-modal data analysis. The data lake Layer accepts, parses and merges multi-modal data, and finally transforms the data in different formats into delta tables. The dynamic index management layer constructs indexes on multi-modal data dynamically according to users’ requests and data characteristics. With the help of data fusion and dynamic index construction, hybrid data exploration is then performed on the transformed multi-modal data. Our framework adopts a typical read/write
decoupling approach. Therefore, write operation (i.e., INSERT, DELETE and UPDATE) about data is only performed in the data lake layer while read operation (i.e., SELECT) is performed in multi-modal data exploration layer. Moreover, all of the aforementioned operations are integrated into Application Programming Interfaces (APIs), enabling users to operate as they wish.

From the implementation point of view, data lake uses Spark for the fusion of multi-modal data, re-framing various data into DataFrame including CSV, JSON, Structured Query Language (SQL), etc. Besides, feature vectors are extracted first by the pre-trained deep learning models. With the embedding of the multi-modal data, indexes for the high-dimensional vectors are generated from the dynamic index management layer. The feature vectors with the indexes are transformed into columns of DataFrame as well. Based on the two layers above, the hybrid data exploration layer is able to do data exploration after the generation of DataFrame for multi-modal data. In this layer, we propose a RESTFUL interface that supports SQL dialects for the convenience of searching and interfaces in GraphX for graph queries.

Heterogeneous multi-modal medical data fusion and management

Due to the variety of source data formats, it is necessary to fuse the multi-modal data before performing queries. In our framework, the fusion relies heavily on the data lake based on Spark. In this section, we first discuss how to implement the fusion process on the structured and unstructured data respectively in Sects. 4.1 and 4.2. We then introduce how to manage and store multi-aspect feature vectors of unstructured data in Sect. 4.3. Next, we propose a request-based method for embedding vectors generation online in Sect. 4.4.

The fusion of structured medical data

There are lots of formats and standards of structured medical data in order to meet different practical demands in various situations. For example, various standards contain SQL, NoSQL, etc.; in terms of file formats, CSV, JSON, XML and others are included. Among all of these diverse data, we aim to find a unified representation model.

We observe that each file contains one identified key and attached features. For join or set operations, the traditional methods process them on individual datasets from different sources separately, which is not efficient. The reason is that all of them are not treated as a whole.

This framework uses a different way to deal with these heterogeneous multi-modal data. Each data file is considered as a set containing different elements. Inside the sets, elements are represented as trees. Every tree is an identified item, and nodes are features or keys. For example, medical data about a patient can be constructed like Fig. 2.

The fusion of multi-source data is represented as the union of different sets. Supposed that we have set A and set B, the target set S of the fusion is represented as:

\[ S = A \cup B \] (1)

Then, the problem of fusing two sets is further decomposed as the merge of two trees with different elements. To merge the elements, we first distinguish two elements by comparing their keys. The algorithm of fusion is shown in Algorithm 1. As a key is either on root node or leaf node, we have to trace back to root node initially for further sub-tree comparison. Three kinds of situations are included: (1) If the keys and all the sub-trees are identical, the two trees are the same and only one is reserved; (2) If the keys are the same while sub-trees are
slightly different, they need further fusion; (3) If the keys are different, both elements are retained.

**Algorithm 1** Fusion of two structured element trees

**Input:** Tree T1, Tree T2

**Output:** Tree T

1. if T1.key1 == T2.key2 then
2.     # Trace back to Root node
3.     if T1.root_node.all_subtrees == T2.root_node.all_subtrees then
4.         T1 == T2
5.     else
6.         T = T1 + (T2 - T1 ∩ T2)
7.     end if
8. else
9.     T = T1 + T2
10. end if
11. return T

The fusion of unstructured medical data

Besides structured data, unstructured data processing is another essential component of our framework. Based on the concepts in Sect. 4.1, unstructured data are regarded as one of the sub-trees in the tree-presented element (See Fig. 3), which means we view unstructured data as one of the features of the elements.

An unstructured data node contains a feature vector node for representation and similarity calculation, a path information node for tracing back to the file, and some other underlying information nodes. The fusion of structured and unstructured data is described in Algorithm 2.

**Algorithm 2** Fusion of structured and unstructured element trees

**Input:** Tree T1, Tree T2

**Output:** Tree T

1. if T1.key1 == T2.key2 then
2.     T = T1 + T2.all_subtrees
3. else
4.     T = T1 + T2
5. end if
6. return T

The storage of multi-aspect feature vectors

In the medical field, we need to deal with many different kinds of unstructured data, such as images, videos, audios, etc. The features of different types of data are different. Even for image data only, images vary differently because there are many medical devices including CT, MRI, etc. The type differences lead to different features. For instance, the feature of gray-scale in an X-ray image is more important than in a line-consisting electrocardiogram in which features of edges will be emphasized.

There is usually more than one feature in a specific unstructured data, such as a medical image, because it is challenging to cover all aspects of information within only one embedding vector. For example, background information, edges information and gray-scale information of an MRI image work together to assist a doctor’s decision. There, using only one vector generated from a single model may lead to the loss of crucial information. To solve the mentioned problem, we propose to use multi-aspect feature vectors generated from different embedding models to represent unstructured data.

To be more specific, for unstructured data in data lake, we use the state-of-the-art models (such as ResNet, Bert, etc.) built in the framework to extract different feature vectors and use them to represent the information of one unstructured data item from different aspects. For hard-to-handle data like videos and audios, we split video data into multi-frame images data and convert audio data into text data, and then use existing computer vision or natural language processing models to embed them.

The feature vector node is demonstrated as Fig. 4. Multi-aspect embedding nodes are attached to the feature vector node. The attached leaf nodes demonstrate information about embedding model name and extracted vector itself.

![Fig. 3 An example of element tree including unstructured data](image-url)
Online generation of embedding
With the development of technology in the medical field and deep learning, medical practitioners may need to extract more features, and use more innovative and advanced models to get more accurate embedding vectors from unstructured medical data. Our framework is scalable to generate embedding vectors for unstructured medical data dynamically.

There are two different situations in that we need to extract new embedding vectors from given unstructured data. The first one is that we can get a more accurate embedding vector to represent the information of one feature by using a more advanced and effective model. With the development of deep learning techniques, many new embedding models have been proposed every year and the feature extraction capability of these models has become stronger and stronger. There, when there is an innovative embedding model that can extract the more accurate feature vector, we can import it into our framework. Then this model can be adopted to generate new embedding vectors for unstructured data to replace the less accurate vectors.

The other situation is that we need to extract a new feature not in the existing features. When doing an exploration task, we may need other features to reflect the information on a specific aspect of the data. According to the advice of experts and doctors, we can utilize corresponding embedding models to generate new embedding for unstructured data like the first situation, and then construct it as a new sub-tree of the data as Fig. 4. For example, when analyzing osteoarthritis diseases, we need to get the features of the images that reflect the severity of the lesion. However, this feature has not been extracted. We can use the corresponding pre-trained embedding model to extract the feature vector of this aspect of the images and save them for data exploration.

Index management of unstructured medical data
Considering the high dimensions of feature vectors from different embedding models for unstructured medical data, it can be very inefficient if we directly use the vectors for similarity calculation. At the same time, it is time-consuming and not convenient for the management of such vectors. Therefore, we propose a dynamic multi-level index management method of unstructured data, using indexes to help improve efficiency. According to different situations, three methods are proposed to meet different requirements: general method, data-driven multi-level method and request-driven multi-level method.

General method
The general method is targeted to meet general requirements with general data sets. In this method, we use a single-dimensional index to construct the order of one particular feature that can best describe the overview of the images.

In the fusion delta table, indexes are generated according to the user’s actual requirements based on the feature vector columns. The way to generate indexes can be chosen by users. This method is straightforward and efficient in the situation when the data set is small and simple.

Data-driven multi-level method
However, the performance of the general method may be affected when the unstructured data are accumulated in a huge amount. We then introduce a data-driven strategy to manage all of the indexes.

Fig. 4 An example of multi-aspect feature vector
The embeddings of images in our framework are multi-aspect as is mentioned in Sect. 4 and multi-dimensional indexes are constructed accordingly. In the data-driven method, we adopt the principle of maximum entropy to build multi-level indexes. For every layer, the feature dividing the data most evenly will be selected to construct the next level index. By dividing evenly, the complexity is significantly reduced when performing queries, thus increasing efficiency.

Suppose there are a large number of images in data lake and these images have two features: image category and shooting time. Divided by image category, 55% images are MRI scans and 45% images are CT scans. Divided by shooting time, 80% images were shot in 2020 and 20% images were shot in 2021. In this case, the first level will be set as image categories rather than shooting time because it can split data more evenly.

**Request-driven multi-level method**

When exploring multi-modal data, we may only be interested in a small class of data with a specific feature. Usually, the level of this feature is too deep, and we need to go through many layers of indexing to find the data with that feature. In this situation, data-driven method might not be efficient. Therefore, we provide a request-driven index management method to improve the efficiency of data exploration.

In Sect. 4.2, we mentioned that for unstructured data, we could extract the required feature vectors. While storing these feature vectors, we can sort these features according to the data exploration behaviors: the most concerning feature is put into the first-level index, the next most concerning feature is put into the second-level index, and so on. The other features are also indexed in this way.

The data lake contains a set of images, including CT scans, MRI scans, etc. Suppose we are performing a data exploration task on osteoarthritis. According to the data-driven approach, the first-level index of these images may be the image category, which is not conducive to efficient filtering and retrieving. Suppose we follow the request-driven strategy, in which the first-level index is dynamically set to the type of disease or body part in the image. In that case, we can directly find all images with osteoarthritis or not (See Fig. 5). And then, we can analyze the images in this set.

Therefore, in the data exploration task, we adopt the request-driven strategy to dynamically generate indexes so that the efficiency can be significantly improved.

**Multi-modal medical data exploration**

Medical data exploration is generally used to assist the medical decision-making process and evaluate the results of medical researches. Meanwhile, the effectiveness of data exploration relies heavily on diverse data structures and various approaches to extract information from different aspects. However, traditional data exploration methods usually extract information from structured data or a single data set, which cannot combine multiple information generated by multi-modal medical data. In this case, we propose hybrid data exploration methods based on data lake, hybrid query (6.1) and graph query (6.2) to process information from multi-modal data and meet the needs of modern medicine. The workflow of multi-modal medical data exploration is presented as Fig. 6. After obtaining tables from data lake and practical requirements from doctors, the exploration process can be divided into two parts, including hybrid query and graph query. Eventually, we get the final matched results as JSON strings which include medical items and cases.
Hybrid query
In some medical scenarios, it is inevitable to use these heterogeneous data including EMRs, CT scans, MRI scans, X-rays, etc. To accurately retrieve patients’ records using heterogeneous data, a hybrid query is executed based on both similarity constraints of feature vectors generated from unstructured data and values of structured data.

Suppose a doctor needs to retrieve similar past cases for references when diagnosing a new patient, noted as A, over 60 years old with MRI scans and CT scans. In this typical situation, we need to search for the target records and retrieve the images according to A’s scan results. Therefore, the situation can be decomposed into two problems: First, the filter condition on structured data about age which should be over 60 years old. Second, the filter condition on unstructured images including MRI scans and CT scans. Feature vectors of A’s scans are query vectors and the scenario requires to find similar images in the database.

Based on our framework, this problem can be settled according to the following steps: First, if needed, new sources of MRI scans and CT scans can be uploaded into our data lake for persistence. The input parameters include upload file path, destination path for persistence, data type, data structured information, data source and data description. The interface will return the first 10 lines of the data uploaded if successful, but failure information if unsuccessful. The interface can manage data persistence in the data lake and deal with semi-structured data like CSV and JSON and unstructured data like text, images and videos. Metadata such as source information are also stored.

Next, data lake provides an interface to extract corresponding feature vectors according to the characteristics of MRI scans and CT scans, and build indexes according to doctors’ requests. In data lake, this step is accomplished with existing techniques in Python and under the supervision of doctors’ knowledge.

Third, we use provided interfaces to merge structured data, semi-structured data and unstructured data, which provide us with a combined table holding all the heterogeneous medical data as in Sect. 4.3. Considering the multi-aspect embedding results, one table would only contain one kind of embedding results in avoidance of information and storage redundancy. Therefore, when there is more than one embedding result, several tables will be created with the indexes.

Then, we are able to realize hybrid query on these tables in the form of SQL dialect. The results returned are hierarchical according to different priorities because of pre-setting in dynamic index construction. In this interface, users can directly use SQL dialect to search on the target delta table with data type and path input. The output is returned in the form of JSON strings.

With the mentioned results returned, several record items under the filter of age and suitable similarity between query feature vectors of MRI scans and CT scans and stored feature vectors from other patients. Doctors can further examine these comprehensive computing outcomes to diagnose and decide further for patients’ treatment. At the same time, the path information recorded along with the feature vectors of these scan images can also help doctors to retrieve back to original scan images beyond getting merely the feature vectors.

Graph query
With the improvement of healthcare, the classification of medical data has become more and more detailed. In this case, we often need to use several characteristics to determine a patient’s specific condition and treatment plan. Therefore, the problem of multiple information matching is worth considering. On the basis of data fusion in data lake, we can transform the data tables into graphs, and also transform the requirements into graphs. Then, we use the interfaces in GraphX to realize the data exploration and obtain the ideal conclusion.

Suppose another medical scenario that a doctor needs to retrieve similar past cases for references when diagnosing a new patient, noted as B. B is a man over 60 years old with X-rays and lots of blood analysis indexes. In this case, we need to execute data exploration according to age, gender, picture information and all blood indicators. We divide the whole process into three parts: First, transform the fusion tables in data lake into graphs. Second, transform the input of multiple pieces of information into a graph. Third, use the interfaces in GraphX to match the input graph with the graph generated by data lake and return similar results. Based on data lake and the whole framework, we can address this problem as follows:

First, as mentioned in Sect. 6.1, according to the prior knowledge of previous cases, we upload tremendous new sources of X-rays, basic patient information and all kind of indexes for detection into our data lake for persistence with the input of file path, destination path, data type and others. At the same time, metadata such as source information of these multi-modal data are stored.

Next, feature vectors are extracted with relevant interfaces according to the needs of doctors and the characteristics of X-rays in data lake. Then we merge them with other structured multi-modal data, such as age, gender, blood pressure and other information, as described in Sect. 4.3. Meanwhile, dynamic storage methods are chosen, and a dynamic index is established and managed based on the individual needs of patient B and data characteristics of X-rays.
After that, the merged results that are presented as tables with many columns and rows are transformed into several big graphs, which we call general graphs. The transformation relies on the interfaces in GraphX, such as the “Array” interface, the “sc.parallelize” interface, and the “Graph” interface. Then, we combine the X-rays and other multi-modal data of patient B in the same way mentioned in the previous steps in data lake, and also transform the combined table into the target graph, which we call target graph B.

Eventually, we are able to perform data exploration with graph query on these tables using the interfaces of GraphX. We consider target graph B as the sub-graph of general graphs and compare them. For example, we use the “filter” interface to input specific parameters and filter out edges and vertices that meet the doctors’ requirements, which represents information from other medical records. In this case, we select cases that are most similar to the target graph B and find differences of the number of edges and other vertices connected to them. Besides, we use the “degrees” interface to calculate the importance of vertices and assess the effectiveness of diverse treatments. After the whole process, several record items and cases will be returned. Doctors can examine these comprehensive computing outcomes to diagnose and decide further precise treatment. Meanwhile, the final step of comparison can be repeated independently by changing different interfaces, which helps doctors to attempt and correct the results more easily.

Implementation and application
Based on the design in the previous sections, we implement our framework on SparkSQL. Section 7.1 describes the prototype implementation of our framework. Section 7.2 introduces the application of our prototype.

Prototype implementation
Based on the tree representation of elements fusion, we use our data lake to realize our framework on SparkSQL platform, which provides rich data structures and powerful computing capability. We focus on the implementation of data fusion.

In SparkSQL, three kinds of data structures are offered: Resilient Distributed Dataset (RDD), DataSet and DataFrame. They are immutable distributed collections of data. In comparison, RDD represents low transformation and control of original datasets. The essence of DataFrame is RDD[ROW]. It organizes data into named columns with more APIs for data manipulation, providing detailed structural information. DataFrame is on a higher level of abstraction and RDD is more suitable for data with a loose structure. Considering the data fusion and implementation of the element tree, the characteristics of DataFrame and the relevant APIs provide us with flexibility during data fusion and exploration queries. The fusion is implemented by the merge function in Spark, which enables us to utilize the calculation capability of this high-performing platform. The fusing algorithm will be further illustrated on data lake in the next paragraph.

The Merge task of Data Lake is divided into two scans. The first scan is an inner join based on the comparison of the key. When the keys are identical, two data rows will be taken out and perform inner join, which is equivalent to Algorithm 1. Then the outer join is performed in the second scan phase between the selected files. Finally, the data rows are added to the output DataFrame.

Considering the two join phases above, merge is a combined optimized execution plan of join in essence. Therefore, based on the computing power of Spark, the queries can be optimized automatically, which provides the fusion process with great convenience and efficiency. Moreover, when executing join on Spark, the SparkSQL platform can automatically generate an optimized implementation plan based on the input file size. The implementation plan includes sort merge join, shuffle hash join and broadcast hash join, thereby improving the efficiency of join.

In the interface, various formats of data (including CSV, JSON, SQL) can be turned into a unified format as DataFrame. Besides, information from unstructured data after feature extraction is stored in JSON files and can be transformed into DataFrame. To complete the fusion process, we input the following parameters: file paths of delta table and JSON file, the primary key of delta table and the primary key of JSON file. Our data lake will then transform the separated files into a merged delta table containing heterogeneous data for further possible read operations and backtracking operations. When any query is required, the merged delta table will be directly read and searched with the user’s SQL request. To record more embedding models, more delta tables will be created and stored for further use.

Application
Our prototype has been applied to Beijing Tsinghua Changgung Hospital to implement heterogeneous data fusion and multi-modal query in knee osteoarthritis disease. Compared to the traditional data system of the hospital, our framework implements knee osteoarthritis data fusion which was previously impossible and has very high data exploration efficiency. The knee osteoarthritis data consists of 156.2 GB of structured data and 1.8 TB of unstructured data which mainly contains patients’ X-ray scans.
We compare our system with the similar system based on Oracle database deployed in that hospital. Oracle processes unstructured data as Oracle LOB. It uses blob to transform scans into binary files with Base64 code and then stores them into clob type of database, based on which traditional indexes are constructed. In comparison, our framework utilizes embedding for compression of X-ray scans in order to construct dynamic indexes and execute dynamic storage with more adaptable strategies. The experiment results of performance comparison between multi-modal data exploration of our framework and the same one based on Oracle show that our data exploration methods outperform the Oracle-based queries by 60% to 70%.

Recently, we have applied multi-modal data query to assist the scientific research in the field of sepsis treatment, we use some customized instruments to regularly collect skin images of patients admitted to ICU, which is completely automated and without human intervention. These collected images will be used to improve the treatment effect of sepsis in clinical research.

In this study, we collect enormous data generated during the treatment of various kinds of patients, most of which are time-series data generated after the patients are admitted to the hospital. The data set contains 63.0 GB of structured data and 67.4TB of unstructured data. To be specific, it includes many vital signs, such as heart rate, mean arterial pressure and temperature; and a variety of test indicators, such as troponin, lymphocyte count and blood potassium; and also some medical behaviors, such as norepinephrine, phenylephrine, fluid intake, SOFA scores and urine volume. Besides, patients’ extremities images are also collected automatically during the whole treatment process.

Based on our computing framework, we can process multi-source heterogeneous data uniformly. We extract feature vectors from these images using some deep learning methods and then build indexes of them to realize the query function of multi-modal data in the next step. Based on this, skin images of patients can be mapped to evaluate the physical condition of sepsis patients from the micro-circulation aspect. The images, together with other vital signs and other test indicators, are used for three-dimensional measurement and rating of patients’ health status, which can be used to study the impact of different medical behaviors at different times in the treatment process on patients’ health status, and also to improve the treatment effect of patients with sepsis.

**Conclusion and future work**

We present a framework for medical multi-modal data based on data lake, implementing heterogeneous multi-modal medical data fusion, index management and hybrid data exploration. Based on the calculation capability and optimized execution plan of SparkSQL, we implement data fusion for multi-modal data and transform them into a merged file for further data exploration. To represent all aspects of multi-modal data, we introduce multi-aspect embedding with deep learning models. For flexibility and efficiency in data exploration, we propose to generate new embedding vectors according to the requirements. Dynamic index management strategies and dynamic storage management selection are also introduced for efficient high-dimensional feature management. Based on the fusion results, we implement hybrid multi-modal data exploration with interfaces in both SQL dialect and GraphX. In future work, we will further develop to add time-series data into our data lake and improve its classification effectiveness [27].

**References**

1. Tong L. Enabling precision medicine by integrating multi-modal biomedical data. Ph.D. thesis, Georgia Institute of Technology. 2020.
2. Zhang Y, et al. Hkgb: An inclusive, extensible, intelligent, semi-auto-constructed knowledge graph framework for healthcare with clinicians’ expertise incorporated. Information Processing & Management. 2020, 56(6): 102324.
3. Cai Q, Wang H, Li Z, Liu X. A survey on multimodal data-driven smart healthcare systems: approaches and applications. IEEE Access. 2019;7:133583–99.
4. He J, Rong J, Sun L, Wang H, Ma J. A framework for cardiac arrhythmia detection from iot-based ecg’s. World Wide Web. 2020, 23:8235–50.
5. Supriya S, Siuly S, Wang H, Zhang Y. Automated epilepsy detection techniques from electroencephalogram signals: a review study. Health Inf Sci Syst. 2020, 8(1):1–15.
6. Gao J, Li P, Chen Z, Zhang J. A survey on deep learning for multimodal data fusion. Neural Comput. 2020, 32(5): 829–64.
7. Chen H, Hu N, Cheng Z, Zhang L, Zhang Y. A deep convolutional neural network based fusion method of two-direction vibration signal data for health state identification of planetary gearboxes. Measurement. 2019;146:268–78.
8. Sarki R, et al. Image preprocessing in classification and identification of diabetic eye diseases. Data Science and Engineering. 2021, 1–17.
9. Sarki R, Ahmed K, Wang H, Zhang Y. Automated detection of mild and multi-class diabetic eye diseases using deep learning. Health Inf Sci Syst. 2020, 8(1):1–9.
10. Li H, Wang Y, Wang H, Zhou B. Multi-window based ensemble learning for classification of imbalanced streaming data. World Wide Web. 2017, 20(6):1507–25.
11. Liang H, Sun X, Sun Y, Gao Y. Text feature extraction based on deep learning: a review. EURASIP J Wirel Commun Netw. 2017, 2017(1):1–12.
12. Sitaula C, Aryal S. Fusion of whole and part features for the classification of histopathological image of breast tissue. Health Inf Sci Syst. 2020;8(1):1–12.

13. Pandey D, Yin X, Wang H, Zhang Y. Accurate vessel segmentation using maximum entropy incorporating line detection and phase-preserving denoising. Comput Vis Image Underst. 2016;155:162–72.

14. Long A, et al. The technology behind tb depot: a novel public analytics platform integrating tuberculosis clinical, genomic, and radiological data for visual and statistical exploration. J Am Med Inform Assoc. 2021;28(1):71–9.

15. Hoffmann K, et al. Integration of mathematical model predictions into routine workflows to support clinical decision making in haematology. BMC Med Inform Decis Mak. 2020;20(1):1–12.

16. Wu J, et al. Hfip: an integrated multi-omics data and knowledge platform for the precision medicine of heart failure. Database. 2021;2021(2021):baab076.

17. Jiang H, Zhou R, Zhang L, Wang H, Zhang Y. Sentence level topic models for associated topics extraction. World Wide Web. 2018;22:2545.

18. Wu P-H, et al. Keyword extraction and structuralization of medical reports. Health Inf Sci Syst. 2020;8(18):1–25.

19. Sun Y, et al. Insma: an integrated system for multimodal data acquisition and analysis in the intensive care unit. J Biomed Inform. 2020;106:103434.

20. Saha SK, Prakash A, Majumder M. “similar query was answered earlier”: processing of patient authored text for retrieving relevant contents from health discussion forum. Health Inf Sci Syst. 2019;7(1):1–9.

21. ALMarwi H, Gharab M, Al-Balta I. A hybrid semantic query expansion approach for Arabic information retrieval. J Big Data. 2020;7(1):1–19.

22. Esposito M, Damiano E, Minutolo A, De Pietro G, Fujita H. Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. Inf Sci. 2020;514:88–105.

23. Du J, Michalska S, Subramani S, Wang H, Zhang Y. Neural attention with character embeddings for hay fever detection from twitter. Health Inf Sci Syst. 2019;7(21):1–7.

24. Dou W, Zhang W, Weng Z, Xia Z. Graph embedding framework based on adversarial and random walk regularization. IEEE Access. 2021;9:1454–64.

25. Hgaish, A. A. Preserve quality medical data toward meaningful data lake by cluster (Preserve quality medical data toward meaningful data lake by cluster, 2019.

26. Kachaoui J, Larioui J, Belangour A. Towards an ontology proposal model in data lake for real-time covid-19 cases prevention. Int J Online Biomed Eng (JIOBE). 2020;16(09):123–36.

27. Supriya S, Siuly S, Wang H, Zhang Y. EEG sleep stages analysis and classification based on weighted complex network features. In: IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, pp. 1–11.

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.