Big Data in Healthcare Institutions: An Architecture Proposal

José Lopes[1], Regina Sousa[2(✉)], António Abelha[2], and José Machado[2]

[1] University of Minho, Gualtar Campus, 4710-057 Braga, Portugal
a82207@alunos.uminho.pt
[2] ALGORITMI Research Center, School of Engineering, University of Minho, Gualtar Campus, 4710-057 Braga, Portugal
regina.sousa@algoritmi.uminho.pt, {abelha,jmac}@di.uminho.pt

Abstract. Healthcare institutions are complex organizations dedicated to providing care to the population. Continuous improvement has made the care provided a factor of excellence in the population, improving people's daily lives and increasing average life expectancy. Even so, the resulting aging has caused patterns to increase day by day and the paradigm of medicine to shift from reaction to prevention. Often, the principle of evidence-based medicine is compromised by lack of evidence on pathogenic mechanisms, risk prediction, lack of resources, and effective therapeutic strategies. This is even more evident in pandemic situations. The current data management tools (centered in a single machine) do not have an ideal behavior for the processing of large amounts of information. This fact combined with the lack of sensitivity for the health area makes it imminent the need to create and implement an architecture that performs this management and processing effectively. In this sense, this paper aims to study the problem of knowledge construction from Big Data in health institutions. The main goal is to present an architecture that deals with the adversities of the big data universe when applied to health.

Keywords: Big Data · Healthcare Information Systems · Real-Time Information System · System Architecture

1 Introduction

Since 1985, with the invention of X-rays, medicine has been intrinsically linked to technology in such a way that it is considered dependent on it [1].

The innovations that have since been sprung up have guaranteed improvements in the quality of life of the community as well as cost reductions in the services provided to it. If, at first sight, everything is beneficial, and social and economic developments increase average life expectancy, the need for investment in health increases accordingly. In view of all this, the philosophy of the institutions has moved from medicine based on the interpretation of symptoms to solving a problem to a medicine that is not only preventive but also predictive [2,3].

R. Hou et al. (Eds.): BDTA 2022, LNICST 480, pp. 297–311, 2023.
https://doi.org/10.1007/978-3-031-33614-0_20

In healthcare institutions, all data are relevant. From the results of laboratory analyses, imaging methods (radiography, magnetic resonance imaging, etc.), to the collection of vital signs through sensors (blood pressure, heart rate, respiratory rate), everything is information with potential knowledge. In short, all data are important and necessary and as a consequence it is increasingly necessary and frequent to collect and store information with interest for the construction of knowledge and consequent evolution. With the evolution of the activity, the data associated with health are generated and aggregated continuously, resulting in large amounts of information in its raw state. This is how the concept of Big Data associated to health emerges [4,5].

The current management tools do not have an ideal behavior for the processing of large amounts of data [6]. Often, the principle of evidence-based medicine is compromised by lack of evidence on pathogenic mechanisms, risk prediction, lack of resources and effective therapeutic strategies. This is even more evident in pandemic situations [7].

Big Data as a research topic has as many opportunities as adversities. Issues such as ambiguity, resistance to change, privacy and information security, reliability and so many others, appear as soon as the subject is discussed. In this way, a robust architecture is essential, so that large-scale organizations, such as health institutions, can demonstrate rigorous evidence based on current techniques and technologies. Only then the progress be believed in a real context of data-driven business.

2 Main Concepts

2.1 Big Data in Healthcare Institutions

Today, Big Data is rooted in various business sectors, yet the health sector is one of those that contains this term, but only flying over all other information systems issues. Health institutions produce large amounts of data, although not everything is in digital format. This makes the processing of this data even more difficult. Besides the diversity of new forms of data (biometric sensors, three-dimensional images, among others), the need arises to digitize data previously recorded on paper [8].

Potential and Challenges of Big Data in Healthcare. With the emergence of several companies interested in the research and use of Large Data in health institutions, several gaps have been defined as priority research needs.

Therefore, the most pointed branches have been:

- The support of research, in a genetic context;
- Transformation of data into information;
- Support for self-care.

However, research on these issues is not always an easy process, much due to the resistance to change present in the nature of health professionals who

have become accustomed to making decisions without the help of any other technology [9]. Moreover, whatever information technology is associated with health, it requires considerable investment with no certainty of return, which makes investment initiatives in this field more scarce than desired [10].

In recent years, the issue of privacy, data security and data ownership has been added to these two challenges, requiring great care with regard to access and processing of third party data [11,12]. Even so, over the years, some goals have been set with regard to the development of information systems. Goals such as increasing the efficiency of healthcare providers, reducing costs and errors, patient-centered and prevention-oriented medicine and personalized medicine have received much attention and investment.

Large data tools have a clear advantage over older technologies and approaches, allowing data analysis, management and processing regardless of format. Therefore, the same tool or processing technology can be used for structured, unstructured or semi-structured data, leading to the construction of relevant knowledge for healthcare institutions and professionals.

Data Sources in Healthcare Institutions. The data sources, in hospital environment, are embedded in the process of electronic health record, medical images, clinical notes or even genetic data [13,14]. Various information can be extracted from the electronic record, such as laboratory results, prescription records, billing data and examination details. In this way, it is necessary to understand the nature, format and use of each type of information:

1. Billing data: Billing data use various codes to document patient symptoms, clinical records and laboratory results;
2. Laboratory data: Laboratory data and vital signs are mostly in structured format. Currently, many dictionaries and various algorithms are developed to reduce complexity of laboratory data;
3. Drug records: Medication records are used to identify the precise characterization of the phenotype. In addition, drug registries are also used to improve disease diagnosis and drug recommendation in the healthcare industry;
4. Clinical notes: Clinical documentation often appears in an unstructured form. Clinical notes are also considered large data and scalable algorithms are used to process such large data;
5. Medical Images: Medical imaging is most often used for diagnosis, planning and therapeutic evaluation. Such data require huge storage space and fast algorithms to process and diagnose diseases. The main challenge with imaging data is that they are not only large in size, but also complex and multidimensional;
6. Documentation of Reports and Tests: Currently, the cost of human genome sequencing is decreasing rapidly with the refinement of high-performance sequencing tools and methods;
7. Sensor devices: Today, more viable medical devices are developed for the continuous monitoring of patient health. These devices generate an enormous amount of health data on an ongoing basis [15].

Data Types in Healthcare Institutions. The data, mainly in health institutions, comes from several sources with quite distinct natures. Thus, they can be classified as Structured, semi-structured or even Unstructured. Excel files, log files, imaging exams (X-rays) are examples of the three types of data, respectively. The following table presents the main differentiating characteristics of the three types of data (Table 1):

Table 1. Differences between structured, semi-structured, and unstructured data. Adapted from [16].

STRUCTURED DATA	SEMI-STRUCTURED DATA	UNSTRUCTURED DATA
Well defined schema	There is not always a scheme	There is no scheme
Regular Structure	Irregular Structure	Irregular Structure
Data independent structure	Embedded Data Structure	Data source-dependent structure
Reduced Structure	Extensive Structure	Extensive Structure
Not evolutive, rigid structure	Very evolutive structure	Very evolutive structure
Closed schemes, integrity restrictions	No associated data scheme	No associated data scheme
Clear distinction in data structure	Distinction in data structure	No distinguish between data structures

2.2 Big Data Engine - Hadoop

Definition. The term Hadoop is one of the most common in today's research technology vocabulary. With a distributed, open source processing structure it manages the processing and storage of data in scalable computer server clusters. To increase the processing capacity of the Hadoop cluster, it is possible to add more servers without having to have expensive memory and CPU resources.

Due to its extreme flexibility it allows organizations to frame and modify the system according to their needs. This technology is the center of an ecosystem of big data technologies that, dealing with various forms of data (structured, unstructured, semi-structured), collect, analyze and manage databases.

Although Hadoop is not considered a data warehouse, it acts as a software structure. Its operation is very briefly based on the distribution of large amounts of data to different nodes, combining the various results later. Therefore, Hadoop provides a high level of durability and availability, which makes it the ideal choice for large loads such as Big Data.

Advantages:

- Scalability: The system can grow easily, just add nodes so you can handle even more data;
- Flexibility: The pre-processing task is no longer necessary. The data can then be stored in its pure state;
- Low cost: The open source framework is free-of-charge and runs on low-cost standard hardware;
- Fault Tolerance: Jobs are always secured since the nodes automatically redirect the work in case of failure;

– Computing Power: Its distributed computing model quickly processes large data, without the components having to have high memory resources, CPU, among others.

Disadvantages:

– Security: Given the various levels of storage and Hadoop network that are unencrypted;
– Vulnerable: The framework is written almost in java, that is heavily exploited by cybercriminals;
– Does not fit for small amounts of data;
– Potential Stability Issues: Since it is an Open source project, there are several programmers working constantly on the project.

Hadoop in Healthcare Institutions. In healthcare institutions in particular, real-time patient monitoring is proportionate to much more detailed supervision of the situation the patient is in and how it has developed since the beginning of the process. Real-time data processing makes it possible to alert the healthcare professional at the immediate moment when the patient's condition changes. Thus, and given that 75/100 of patients' data, present in healthcare institutions, is not structured in nature, a careful analysis of this data in real time can not only eliminate and/or incorrectly classify exams and wrong diagnoses, but also decrease the costs of the healthcare institution.

The term real time is often associated as something instantaneous, but there are deadlines for data entry and exit that are variable from institution to institution. Ideally, the system should have the ability to receive data in a continuous flow, producing and sending results in a matter of milliseconds. However, in some cases (e.g., a wind turbine), if the response is produced in a period of one minute or even two, this processing is also considered real time, since there is a whole mechanical process from the sensor in the turbine to the system that is receiving the data.

Most healthcare institutions are still investigating the best tools for Big Data treatment in order to improve patient care. Hadoop is in fact a very viable solution that tries to address some challenges in the hospital environment such as electronic health records (EHR) as these have too many free text fields for clinical notes.

There are several advantages of using Hadoop in healthcare institutions. As has been mentioned, currently much of the information is in the form of unstructured data. This category includes medical notes, laboratory reports, imaging examinations, physiological signs, among others. In addition, the average data storage capacity in healthcare facilities is three days of data per patient, which seriously limits the opportunity for data analysis. Therefore, Hadoop allows the storage and availability of data without the cost of these tasks being astronomical. In addition, this tool can also serve as a data organizer and analyst.

3 Proposed Architecture

This proposal seeks to create an open-source architecture that is suited for the capture and real-time processing of large volumes of information within the healthcare area, which when later analyzed are valuable for the acquisition of useful outcomes that provide solutions to support health professionals in the decision-making process.

Moreover, through various big data sources, the aim is to develop a clinical decision support system based on real-time knowledge discovery, while maintaining reliability and availability, as well as allowing easy adaptation in terms of scalability.

The proposed architecture is illustrated in Fig. 1 and can be divided into five layers: data layer, data aggregation layer, data analytics layer, data security and information exploration.

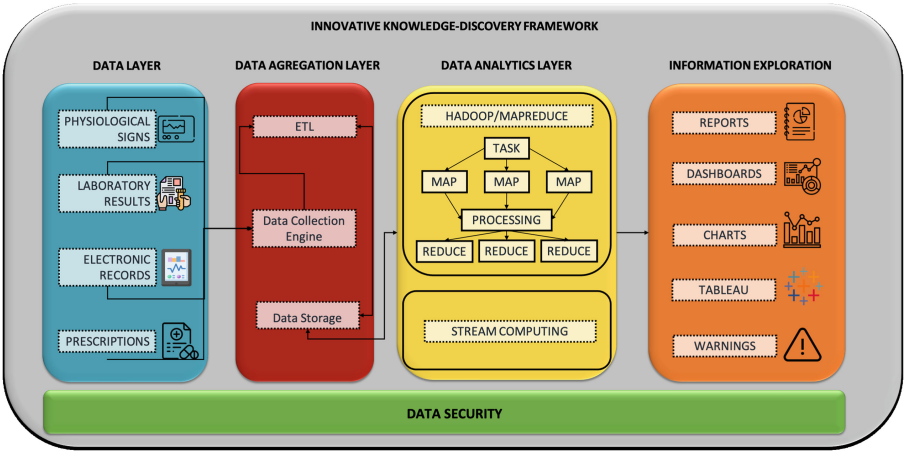

Fig. 1. Proposed Architecture.

3.1 Structure

The next step is to understand each layer of this framework, along with its associated components.

Data Layer. The proposed architecture takes advantage of high volumes of data from numerous healthcare sources, mainly physiological signs, laboratory results, electronic records, and prescriptions. The data's composition may differ due to their structured/unstructured nature, so this architecture must have the ability to collect data in all its formats.

Data Aggregation Layer. This layer concerns the collection, incorporation, transformation, and storage of data. This is an important step since the accuracy of results from data analysis depends heavily on the amount and quality of the exploited data. Therefore, it is important to gather high-quality, accurate data in a large enough amount to create relevant results.

Data Collection
The data collection engine decides how the data is collected from several data sources in order to build the data pipeline. Thus, data collection is where the extraction portion of the ETL process is performed.

At the initial collection stage, initial metadata can be generated, which facilitates subsequent aggregation or look-up methods.

ETL Process
An Extract-Transform-Load (ETL) process is an ordered sequence of operations that seeks to systematically process data so as to make it available in a more accessible format.

The implementation of an ETL process focuses on the following tasks:

1. Extract: Data extraction from different sources;
2. Transform: The extracted data is converted into the user's required format. For processing and validating the data, it is required some sort of data transformations, such as data cleansing, filtering, standardization, flow validation, joining, splitting, sorting, among others;
3. Load: In the final stage of the ETL process, the processed data is loaded into the destination, usually a database or a distributed file system [17].

Data Storage
Storage of Big Data is an important issue for its management. Management is only possible when the storage of data is done properly and efficiently such that the retrieval of data from huge datasets is simpler and user-friendly.

Data for batch processing is normally stored in a distributed file store that can maintain high volumes of large files in various formats.

Data Analytics Layer. Here the aim is to process and examine all the collected data in order to uncover meaningful patterns, relations and other insights.

With the currently available technology, it is possible to analyse data in real-time, through stream computing.

Information Exploration. In the final layer, the focus is on gathering data value and extract business intelligence, providing an ergonomic and user-friendly representation of the data-driven outcomes, in terms of graphs, tables and/or images.

Components

Apache Kafka
Apache Kafka is a distributed high-throughput platform for a real-time environment using a publish-subscribe messaging system, used for high-performance data pipelines, stream processing, log aggregation, and data integration [18].

Advantages:

- Low Latency: Handles messages with a low latency value (on the millisecond's scale), even when handling a large number of messages;
- High-Throughput: Due to low latency, Kafka can manage high-velocity and high-volume data;
- Fault-Tolerance: Provides resistance to node/machine failure within a cluster;
- Distributed: Contains a distributed architecture which makes it scalable. Uses capabilities such as partitioning and replication;
- Scalability: Handles a large number of messages simultaneously. Kafka can be scaled-out by adding additional nodes without incurring any downtime;
- Durability: Presents a replication feature, which makes data/messages persist on the cluster;
- High Concurrency: Kafka can deal with thousands of messages per second in low latency conditions with high throughput, allowing the reading and writing of messages into it at high concurrency;
- Easily accessible: Since all data gets stored in Kafka, it becomes easily accessible;
- Consumer Friendly: Allows producers and consumers to operate independently and at separate times. Moreover, Kafka can integrate well with a variety of consumers written in a variety of languages;
- Real-Time Handling: Able to handle real-time data pipelines;
- All the data that a producer writes go through Kafka. Therefore, a single integration is sufficient to automatically integrate with each producing and consuming system;
- Kafka allows data/messages to integrate directly into applications using APIs, providing additional features and functionalities.

Disadvantages:

- Kafka does not contain a complete set of management and monitoring tools;
- Kafka's high performance depends on whether the message requires additional processing. It can perform well if the message is unchanged because it uses the capabilities of the system, but its performance is significantly reduced if additional processing is required;
- Kafka only matches with the exact topic's name, not supporting wildcard topic selection. This is related to the fact that selecting wildcard topics makes it incapable to address certain use cases;
- As the messages' size increase, brokers and consumers start compressing them, increasing memory usage. After decompressing the data flow, the node memory gets slowly used. This compressing and decompressing of messages affects its performance and throughput;

– When the number of queues in a cluster increases, Kafka's performance decreases [19].

Talend
Talend Open Studio is an open-source project that supports ETL-oriented implementations and is provided for on-premises deployment as well as in a Software-as-a-Service (SaaS) delivery model [17]. Talend Open Studio for Big Data helps to develop faster with a drag-and-drop User Interface (UI) and pre-built connectors and components [20].

Advantages:

– It is a very flexible, scalable and performance-driven solution for executing data manipulation and extraction on Big Data;
– Since it is open-source, it enables users to access, transform, move and synchronize data through big data components such as Hadoop, Hive, Kafka, Spark and NoSQL databases (e.g. HBase), making them easier to use;
– Customizes and creates components and code to extend a project. It can generate both MapReduce and Java code;
– The user usually just needs to drag and drop the components and configure a few parameters;
– Connects with several big data distributions like Apache, MapR, Cloudera, HortonWorks, and other cloud solution providers [20,21].

HDFS
In the Hadoop ecosystem, the Hadoop Distributed File System (HDFS), as its designation suggests, is a distributed and scalable file system and the primary data storage system used by Hadoop applications [22].

Advantages:

– Large data storage: stores a variety of data of any size and format;
– Provides scalable and fast data access;
– Can store files across multiple machines;
– Cost effectiveness: relies on much less expensive commodity storage disks and can be implemented on low cost and easily replaceable hardware;
– Fast recovery from hardware failure: HDFS is highly fault-tolerant and can automatically recover on its own;
– Streaming data access: HDFS is built for high data throughput, which is best for access to streaming data.
– Shares its hardware with MapReduce. Thus, HDFS is optimized for MapReduce workloads and provides high performance for sequential reads and writes [23].

MapReduce
MapReduce allows the scalability of countless servers (nodes) in a Hadoop cluster. Its purpose is to divide the collected data in order to be analysed into smaller

and independent parts, and to be processed in parallel, reducing the processing time [22].

Advantages:

- Scalability: MapReduce can scale across thousands of nodes;
- Flexibility to process different types of data from various sources;
- Memory requirements: MapReduce does not require large memory, meaning that it can work with a minimal amount of memory and still quickly produce results;
- Fast: MapReduce is located in the same servers as HDFS, which allows for faster data processing;
- Parallel processing: Tasks are divided in a way that allows their execution in parallel. Parallel processing allows multiple processors to take on these divided tasks, which helps them run programs in less time;
- Availability: MapReduce processes data by sending it to an individual node, as well as forwarding the same set of data to other nodes in the network. Therefore, in case of failure in a particular node, other copies are available and can be accessed whenever the need arises;
- Fault Tolerance: MapReduce has the ability to quickly recognize faults and then apply a quick and automatic recovery solution;
- Cost-effective: Due to its high scalability, MapReduce reduces the cost of storage and processing in order to meet the growing data requirements;
- Security and Authentication: MapReduce works with HDFS security that allows only approved users to operate on data stored in the system [24].

Tableau
The Tableau platform is a usual choice for modern business intelligence and is known for rapidly turning data into useful insights, helpful in the decision-making process.

Advantages:

- Ease of use;
- High performance;
- Multiple data source connections;
- Remarkable visualization capabilities.

Disadvantages:

- High cost and inflexible pricing;
- Limited data preprocessing;
- Time- and resource-intensive staff training [25,26].

Monitoring and Alerting. Modern-day's information systems require real-time performance controls for a better understanding and interpretation of data.

The monitoring process helps to supervise the data flow in a system. It allows users to find complications before they develop into problems and helps

to maintain high availability and quality of service. Therefore, monitoring aims to identify faults and assist in their elimination, providing helpful assistance in the decision-making process.

Alerting is somewhat allied to the monitoring process. It involves the capability of a monitoring system to detect and notify users about abrupt changes in the system's state. Thus, alerting is essentially an automated way to send notifications to users when there is an immediate problem in the system [27].

In the proposed architecture, the monitoring process is done through the conception of reports, where we organize the retrieved data into visual indicators such as dashboards, charts, tables, and time-series, in order to keep track of its performance. Since we are dealing with data originated from healthcare institutions, it is important to be aware of what is happening with the extracted information in real-time. Thus, reporting helps to alert, in the form of warnings or notifications, when the data falls outside of expected ranges (or some other pre-defined criteria), inducing us to react and take appropriate action, as necessary. For example, if a patient's blood pressure increases alarmingly, the system will send a warning in real-time to the designated doctor who will then take measures to lower the patient's blood pressure.

Data Security. Tied to the vital activities to manage the access to the system's data and services, Security involves:

1. Authentication: Verify and validate the identity of a user or service, so that only legitimate users have access to the data and services.
   - In Hadoop, it is common to authenticate with Kerberos and use LDAP as a backend. Depending on the Hadoop component, other authentication possibilities are available, such as SAML, OAuth and classic HTTP authentication;
   - Apache Kafka's authentication is done with SSL or SASL.
2. Authorization: Ensures that the user or process has the rights to access resources or services. In other words, it provides permission to the user (or process) whether he can access the data or not.
   - In Hadoop, access to HDFS is managed through UNIX access rights where users are assigned to user groups;
   - On other Hadoop components, such as Apache Kafka, authorization is controlled by role-based Access Control Lists (ACLs), which include features such as regular expressions for pattern matching.
3. Data Protection: Involves the use of techniques such as encryption and data masking, vital in preventing sensitive data access by unauthorized users and applications.
   - Hadoop provides encryption for HDFS. Thus, since several Hadoop components write their temporary files to HDFS, those files will automatically be secured. This HDFS-level encryption allows applications to run transparently on the encrypted data, while also preventing filesystem or OS-level attacks;

  - In Apache Kafka, data is encrypted using SSL/TLS, which keeps data encrypted between the producers and Kafka, as well as the consumers and Kafka;
  - In some cases, if data encryption is not required by law or business reasons, partial encryption of data can be done, mainly beneficial for performance purposes. For example, MapReduce uses format-preserving encryption and masking techniques, facilitating faster analytical processing between applications.
4. Auditing: Keeping track of events that happen within the system to support forensic analysis in the event of a breach or corruption of data. Auditing also monitors the activity of an authenticated and authorized user or process, including what data was accessed, added, and modified.
  - Each of the Hadoop components offers audit capabilities to ensure that the users and administrator's activities can be logged;
  - Every component of a big data platform allows logging, either to the local file system or into HDFS [28,29].

3.2 Hands-On: Proposed Process

After an extensive description of each specific layer and its associated tools, this section will focus on the clarification of the exact data flow process throughout the entire architecture. This proposal involves a hybrid architecture that seeks to build an ecosystem capable of supporting both batch and stream processing, combining similar or related components and APIs, and visualization of information in real-time.

Firstly, we start by presenting the different data sources available. In this case, the ones accessible in healthcare institutions. Thereby, the beginning of this process is characterized by the capture of data derived from these healthcare data sources.

To deliver real-time ingestion, Apache Kafka is connected to these data. This connection to Kafka can be accomplished by different approaches: the usual recommendation, which is to use the APIs developed by the Apache Software Foundation, or to use additional technology to facilitate its handling.

Furthermore, Kafka will be used for both data ingestion and stream processing. Since we want a real-time (or near real-time) architecture, the low latency provided by Kafka, meaning that it is optimized to deal with a very high volume of data messages with minimal delay, allows real-time ingestion and real-time processing with time frames shorter than milliseconds. Also, its easy configuration, fault tolerance, high scalability, and the fact that a single tool will be used to collect and process data, makes a Kafka ingestion - Kafka streaming connection an ideal choice for this proposal.

However, we cannot perform ETL transformations in Kafka. Thus, for an ETL-oriented implementation, Kafka's data can then be integrated into Talend Open Studio through its standard connectors and components. Talend's graphical design environment enables the exporting and execution of standalone jobs in runtime environments. Following Talend's connection to an existing Kafka topic,

we can persist data in the data object and perform end-to-end ETL transformations. Afterwards, Talend can create a job that writes these data in an HDFS (it can also access data that is stored in an HDFS if necessary).

Alternatively to Kafka's stream processing, we use Hadoop to perform batch processing. After the transformed data's storage in an HDFS, we use it in the analysis layer for batch processing. In batch processing, batches undergo processing through MapReduce, which can later be, once more, stored in an HDFS.

Lastly, following the data's processing and analysis, it is possible to consume and explore the acquired information, where it can be displayed in a more understandable and intuitive form. To do so, we will use Tableau. Tableau connects to Apache Kafka and, through this connection, it is possible to obtain near real-time results, which will be continuously updated in dashboards as new data is ingested and processed. Despite this, and due to Tableau's higher cost, it is advantageous to make use of another tool. So, while Tableau constructs evaluation parameters, we will additionally make our own evaluation, where we will view these parameters and manipulate them, creating, for example, warnings based on the acquired information from Tableau.

4 Discussion and Conclusions

The proposed architecture is capable of supporting healthcare analytics by providing batch and real-time results, combined with an extensible storage technology solution, which will benefit healthcare institutions to a great extent. In fact, its main advantage is the ability to combine historical data with real-time data and produce results in real-time.

Moreover, the insertion of a stream processing tool is beneficial for handling healthcare data that require an almost immediate response. In doing so, the data processing framework achieves low latency, while also presenting in-memory computing, indicating that nearly all processing is performed in the cluster's memory and only the final output is stored on a storage disk.

Throughout the paper, several tools were introduced and briefly clarified, followed by their specific insertion in each layer of the architecture. Hadoop's MapReduce was employed for batch streaming, performed on data stored in HDFS, while Apache Kafka was implemented for stream processing.

However, since historical data is extensive and new information is constantly produced, it can be difficult to maintain the production of real-time results. Thus, this architecture carries a substantial implementational complexity. Also, since batch and stream processing use different technologies, they need to be developed separately, requiring higher efforts in the architecture's implementation.

It is also important to denote that other open-source tools are available. Regarding data collection tools, Apache Flume and Apache NiFi are some usual choices. As for stream processing systems, Apache Spark, Apache Flink, and Apache Storm are often employed.

Since this paper mainly focuses on the theoretical side of the architecture, offering a possible architecture for dealing with Big Data in healthcare institutions, future work primarily concerns its practical implementation, so as to

evaluate its performance and establish how its execution is beneficial when dealing with huge amounts of healthcare data.

Acknowledgments. This work is funded by "FCT-Fundação para a Ciência e Tecnologia" within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Editors, H.: Scientist discovers X-rays (2019)
2. Barnes, T.J.: Big data, little history. Dialogues Hum. Geogr. **3**(3), 297–302 (2013)
3. Sivarajah, U., Kamal, M.M., Irani, Z., Weer-akkody, V.: Critical analysis of big data challenges and analytical methods. J. Bus. Res. **70**, 263–286 (2017)
4. Ramage-Morin, P.L.: Successful aging in health care institutions. Statistics Canada (2005)
5. Neves, J., et al.: A deep-big data approach to health care in the AI age. Mob. Netw. Appl. **23**(4), 1123–1128 (2018)
6. Neto, C., Brito, M., Lopes, V., Peixoto, H., Abelha, A., Machado, J.: Application of data mining for the prediction of mortality and occurrence of complications for gastric cancer patients. Entropy **21**(12), 1163 (2019)
7. Wang, Y., Kung, L., Byrd, T.A.: Big data analytics: understanding its capabilities and potential benefits for healthcare organizations. Technol. Forecast. Soc. Chang. **126**, 3–13 (2018)
8. Feldman, B., Martin, E.M., Skotnes, T.: Big data in healthcare hype and hope. Dr. Bonnie **360**, 122–125 (2012)
9. Pereira, A., et al.: Improving quality of medical service with mobile health software. Procedia Comput. Sci. **63**, 292–299 (2015)
10. Bates, D.W., Saria, S., Ohno-Machado, L., Shah, A., Escobar, G.: Big data in health care: using analytics to identify and manage high-risk and high-cost patients. Health Aff. **33**(7), 1123–1131 (2014)
11. Raghupathi, W., Raghupathi, V.: Big data analytics in healthcare: promise and potential. Health Inf. Sci. Syst. **2**(1), 3 (2014)
12. Kruse, C.S., Goswamy, R., Raval, Y.J., Marawi, S.: Challenges and opportunities of big data in health care: a systematic review. JMIR Med. Inform. **4**(4), e38 (2016)
13. Neto, C., et al.: Different scenarios for the prediction of hospital readmission of diabetic patients. J. Med. Syst. **45**(1), 1–9 (2021). https://doi.org/10.1007/s10916-020-01686-4
14. Martins, B., Ferreira, D., Neto, C., Abelha, A., Machado, J.: Data mining for cardiovascular disease prediction. J. Med. Syst. **45**(1), 1–8 (2021). https://doi.org/10.1007/s10916-020-01682-8
15. Manogaran, G., Thota, C., Lopez, D., Vijayakumar, V., Abbas, K.M., Sundarsekar, R.: Big data knowledge system in healthcare. In: Bhatt, C., Dey, N., Ashour, A.S. (eds.) Internet of Things and Big Data Technologies for Next Generation Healthcare. SBD, vol. 23, pp. 133–157. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49736-5_7
16. Li, G., Ooi, B. C., Feng, J., Wang, J., Zhou, L.: Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 903–914 (2008)

17. Sreemathy, J., Nisha, S., Prabha, C., RM, G.P.: Data integration in ETL using Talend. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1444–1448. IEEE (2020)
18. The Apache Software Foundation. Apache Kafka Documentation. https://kafka.apache.org/intro. Accessed 11 Mar 2021
19. DataFlair: Advantages and Disadvantages of Kafka (2019). https://data-flair.training/blogs/advantages-and-disadvantages-of-kafka/. Accessed 12 Mar 2021
20. Talend. Talend Open Studio for Big Data. https://www.talend.com/products/big-data/big-data-open-studio/. Accessed 28 Mar 2021
21. Katragadda, R., Tirumala, S.S., Nandigam, D.: ETL tools for data warehousing: an empirical study of open source Talend Studio versus Microsoft SSIS (2015)
22. White, T.: Hadoop: The Definitive Guide. O'Reilly Media Inc (2012)
23. TutorialsCampus. HDFS Overview. https://www.tutorialscampus.com/hadoop/hdfs-overview.htm. Accessed 29 Mar 2021
24. Tutorialspoint: Advantages of Hadoop MapReduce Programming. https://www.tutorialspoint.com/advantages-of-hadoop-mapreduce-programming. Accessed 29 Mar 2021
25. Tableau. Business Intelligence and Analytics Software. https://www.tableau.com/why-tableau. Accessed 14 Mar 2021
26. SaM Solutions: Pros and Cons of Tableau Software for Data Visualization (2017). https://www.sam-solutions.com/blog/tableau-software-review-pros-and-cons-of-a-bi-solution-for-data-visualization/. Accessed 14 Mar 2021
27. Ligus, S.: Effective Monitoring and Alerting. O'Reilly Media Inc. (2012)
28. Chang, W.L., Boyd, D., Levin, O.: NIST big data interoperability framework: volume 6, reference architecture. National Institute of Standards and Technology (2019)
29. Deloitte Consulting GmbH: Five key principles to secure the enterprise Big Data platform (2017)