

Apache Spark for Analysis of Electronic Health Records: A Case Study of Diabetes Management



Kanhaiya Sharma^{ID}, Deepak Parashar^{*ID}, Om Mengshetti^{ID}, Raasha Ahmad^{ID}, Rewaa Mital^{ID}, Perna Singh^{ID}, Muskan Thawani^{ID}

Symbiosis Institute of Technology Pune, Symbiosis International (Deemed) University, Pune 412115, India

Corresponding Author Email: parashar.deepak08@gmail.com

Copyright: ©2023 IIETA. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>)

<https://doi.org/10.18280/ria.370616>

ABSTRACT

Received: 12 August 2023

Revised: 1 September 2023

Accepted: 7 October 2023

Available online: 27 December 2023

Keywords:

diabetes, machine learning, electronic health records, Apache Spark, feature selection

Electronic Health Records (EHRs), heralded for their potential to revolutionize healthcare outcomes, function as repositories for invaluable data. This study offers a compelling exploration into the integration of Apache Spark for EHR analysis, with a specific focus on elevating diabetes care. Leveraging Apache Spark alongside a robust machine learning framework, we automated EHR analysis by processing extensive datasets, conducting thorough preprocessing, and extracting pertinent features. The inherent distributed processing capabilities of Apache Spark facilitated concurrent training and evaluation of machine learning models. Its in-memory data processing markedly reduced reliance on disk input/output, thereby enhancing performance and scalability. This methodology enabled swift and thorough EHR data analysis, with ensuing insights effectively visualized and reported. This empowered healthcare professionals to make informed decisions. The iterative nature of the process allowed for continuous refinement, enhancing healthcare outcomes based on insightful data. The synergy between Apache Spark and machine learning techniques in EHR analysis emerged as a potent and efficient strategy. This approach exhibits promise in significantly advancing healthcare outcomes by enabling effective prediction and management of diabetes, ultimately contributing to superior patient care and reducing healthcare costs. The findings underscore the transformative potential of integrating contemporary data analysis tools within the healthcare sector.

1. INTRODUCTION

In recent times, various applications generate a vast amount of data that accumulates over time. Extracting relevant information from these datasets can be challenging due to their massive size. Data mining has emerged as an effective method to obtain a higher level of knowledge amidst such large volumes of data. Oneway to gather information from diabetic patients is to use WBANs to collect individuals' biomarkers and send them to a base station using a set of body [1]. Therefore, data mining is an important technique that can be used to extract useful insights from large datasets.

Diabetes is a chronic condition that affects millions of individuals globally and is becoming more common. Millions of individuals worldwide are thought to have diabetes, and the number is thought to be rising [2]. For the purpose of reducing complications and enhancing outcomes, it is imperative to identify those who are at high risk for developing diabetes early on. A wealth of data, including demographic data, clinical data, and test findings, can be found in electronic health records (EHRs) that can be used to predict the consequences of diabetes. The quantity, complexity, and variability of EHRs, however, make processing and analysis difficult. An effective and scalable platform for processing and analyzing big data is offered by Apache Spark, a distributed

computing framework. This paper presents a case study of Apache Spark being used to analyze EHRs and enhance diabetes treatment. The study's goals and contributions are as follows. Analyze Apache Spark's effect on the speed at which machine learning techniques can be used to EHR analysis. Examine Apache Spark's potential as a formidable platform for increasing EHR analysis in healthcare to improve patient outcomes and diabetes management.

The remainder of this study is divided into the following sections. We provide a thorough context and literature review in Section II. Section III highlights the need for big data analytics. Section IV details the suggested solution. Section V contains the conclusion and the future scope.

1.1 Apache Spark

Big dataset management and analysis are made possible by Apache Spark, a distributed computing framework. It was developed by and distributed in 2010 by the Apache Software Foundation. Scala, a programming language that is compatible with the Java Virtual Machine, was used to create the open-source project Spark. Spark has gained popularity among researchers and data scientists since its release as a result of its capacity to handle massive datasets quickly and effectively. Spark comes with a number of advantageous features, such as

in-memory processing, which improves efficiency by caching data rather than writing it to disk after each operation, hence minimizing disk I/O. It is a strong data analysis tool because of its adaptability in handling organized, semi-structured, and unstructured data. The parallel processing feature distributes tasks across a computer cluster using a master-slave architecture, enabling efficient processing of large datasets. Additionally, Spark offers rich libraries for machine learning, graph processing, and streaming data analysis. These libraries are designed to work seamlessly with Spark's core processing engine, providing researchers and data scientists with a comprehensive set of tools for data analysis. Standalone mode of Apache Spark Architecture is shown in Figure 1.

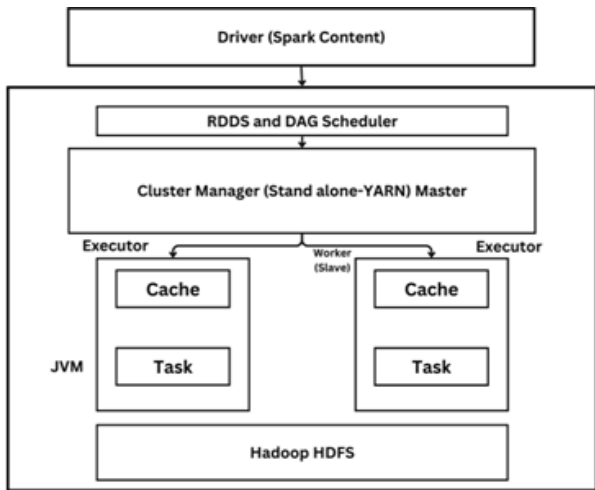


Figure 1. Standalone mode of Apache Spark architecture

1.2 Diabetes

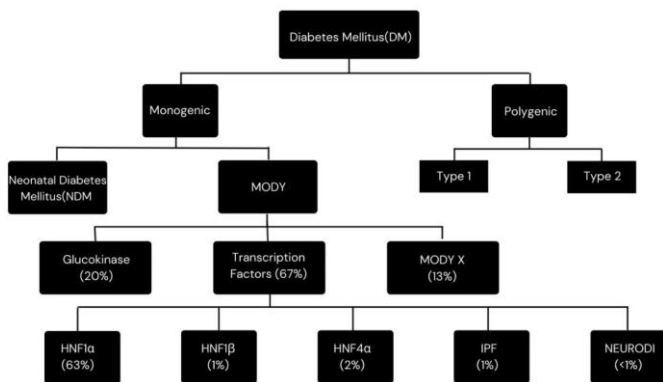


Figure 2. Types of diabetes and its bifurcation

Over 30 million individuals in the US alone suffer from diabetes, a chronic illness. To avoid consequences including cardiovascular disease, renal damage, and blindness, diabetes must be effectively managed. EHRs contain a wealth of information on diabetes management, including laboratory values, medication use, and clinical outcomes. Analyzing this data can help identify patient characteristics and clinical factors associated with successful diabetes management. Types of diabetes and its bifurcation is provided in Figure 2. Diabetes is a prevalent medical condition affecting millions worldwide, encompassing various types such as Type 1, Type 2, gestational diabetes, and rare variants. Type 1 diabetes results from the immune system's attack on insulin-producing pancreatic cells. Type 2 diabetes arises due to insufficient

insulin production or resistance. Gestational diabetes typically resolves post-pregnancy. Uncommon forms include monogenic diabetes, diabetes linked to cystic fibrosis, and drug-induced diabetes, each characterized by distinct symptoms, risk factors, and treatment options.

2. LITERATURE REVIEW

The use of big data analytics in the healthcare sector has increased recently, with Apache Spark engine as a key tool for processing and analyzing enormous amounts of healthcare data. Because they hold a variety of patient data, including demographics, diagnoses, prescriptions, and treatments, electronic health records (EHRs) in particular have emerged as a vital source of information for medical professionals. The study on the usage of Spark for EHR analysis is reviewed in this section [3]. Many research has employed Apache Spark for EHR analysis, proving how well it can handle vast amounts of medical data. Based on EHR data, a predictive model for hospital readmissions was created using Apache Spark. The study found that Apache Spark was able to handle the large volume of data and process it in real-time, resulting in accurate predictions of readmission risk [4]. For analysis of diseases-specific data, such as diabetes management, other research has employed Apache Spark. A model for predicting diabetic complications based on EHR data was created using Apache Spark. The study's discovery that the model could correctly forecast difficulties illustrated Apache Spark's promise for customized medicine [5]. Apache Spark has been utilized for EHR data clustering and categorization in addition to predictive modeling. To enable targeted interventions and better care management, Apache Spark is used to group patients based on how they utilize healthcare services [6].

The processing of healthcare data using Apache Spark has been found to be more effective and scalable when compared to other big data analytics solutions like Hadoop and Map Reduce [7]. Overall, the research points to Apache Spark as a potential tool for EHR analysis, offering scalable and effective processing capabilities for huge amounts of medical data. Future studies are required to examine Apache Spark's potential in customized medicine.

The early detection and diagnosis of diabetes utilizing machine learning techniques and Apache Spark, a large data processing framework, is the issue this study attempts to solve. The goal is to develop a trustworthy binary classification model that can assess a patient's risk for diabetes based on crucial clinical factors such age, gender, BMI, blood pressure, glucose level, and other pertinent information [8]. The objective of the study is to evaluate the effectiveness of several machine learning algorithms, including support vector machines, decision trees, and logistic regression, and to determine the most effective feature engineering and model optimization techniques. The project will also investigate real-time deployment alternatives for the model using Apache Spark. The results of this study can contribute to improving diabetes detection and diagnosis, thereby reducing the risk of complications [9]. Recent years have witnessed a surge in healthcare's use of Apache Spark for big data analytics. Electronic Health Records (EHRs) have emerged as a vital source of patient data. Multiple studies have shown Spark's effectiveness in handling vast healthcare datasets, from predicting hospital readmissions to diabetes complications. Comparisons with other tools consistently favor Spark's

efficiency. Current research explores Spark's potential in early diabetes detection, aiming to enhance diagnosis and patient outcomes. Key research gaps include expanding data sources to encompass diverse healthcare data, addressing interoperability challenges, implementing real-time monitoring, managing ethical and privacy issues, deploying machine learning models in clinical practice, enhancing patient engagement, evaluating long-term impacts, optimizing Apache Spark's performance, and integrating it with telemedicine platforms.

3. NEED FOR BIG DATA ANALYTICS

3.1 Current need

Coping with Expanding EHR Data: The increasing complexity and volume of electronic health records (EHRs) necessitate big data analytics, specifically Apache Spark, to efficiently store, manage, and analyze these vast datasets [10]. **Real-Time Healthcare Decision-Making:** Apache Spark's real-time processing capability is crucial for timely healthcare decisions and interventions. This addresses the urgent need for scalable and efficient EHR data processing in the face of growing data complexity and volume.

3.2 Future need

Advancing Personalized Medicine: In the future, Apache Spark will be increasingly essential in EHR analysis to support personalized medicine. It will enable healthcare providers to harness patient data from EHRs for precise clinical decision-making, enhancing individualized patient care [11]. **Enhancing Population Health Management:** Apache Spark's efficient processing of EHR data will play a pivotal role in population health management, allowing healthcare providers to draw insights from large datasets to make more informed and effective interventions, ultimately improving healthcare outcomes on a broader scale. Additionally, Apache Spark can be used to find patterns, risk factors, and potential actions to enhance results, analyze data on population health. On a larger scale. Therefore, the future need for Apache Spark in EHR analysis is to provide healthcare providers with the tools to leverage patient data for personalized medicine and population health management.

4. PROPOSED SOLUTION

The proposed solution design philosophy through a data flow diagram is shown in Figure 3. Millions of individuals throughout the world suffer from the chronic condition of diabetes. It happens when the body is unable to control blood sugar levels, which can result in a number of health issues. Early diagnosis and management of diabetes are essential for minimizing complications and improving health outcomes. In recent years, diabetes detection and prediction using machine learning algorithms has showed promise. In this article, we will explore how to use machine learning for diabetes detection using Apache Spark.

The suggested strategy makes advantage of learning methods like DT and RF, which offer high performance and speed. To categorize people into two groups, normal and abnormal, machine learning has developed a classification

model. This model may be examined based on evaluation indices like accuracy and runtime and tested using test data. The proposed method can be further explained using the Apache Spark architecture. This method involves converting the dataset of diabetes patients into the storage-friendly Resilient Distributed Dataset (RDD) format. RDD is a method of data storage in HDFS or the Apache Spark distributed memory architecture. This approach uses a number of compute node clusters that are split into primary and secondary classes. The main cluster is in charge of overseeing the sub-clusters and allocating the computational workload, including Machine Learning, to other Spark nodes. By dispersing data about sick or healthy persons in the Apache Spark Distribution System, machine learning techniques may be used to the data existing there with the help of mapping operations. Using Apache Spark to reduce runtime. Future research into integrating WBAN and Apache Spark technologies to evaluate patient status in hospitals may be undertaken. It is crucial to remember that the suggested framework is only intended to be used for learning, and that Apache Spark can be used to explain the specifics of the suggested method. Using this technique, diabetes patients' data may be analyzed and divided into two groups. The suggested method offers a quick and effective solution to analyze big datasets by utilizing distributed computing and the RDD format. Overall, the suggested approach is a potent tool for data analysis with potential applications in many areas.

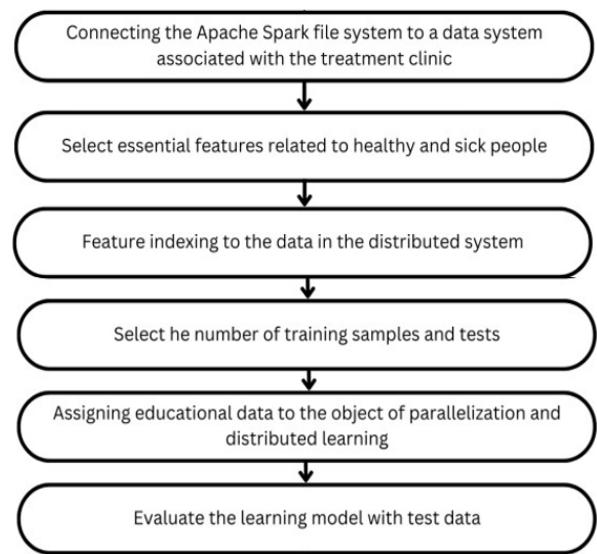


Figure 3. Proposed solution design philosophy through a data flow diagram

4.1 Data collection and preprocessing

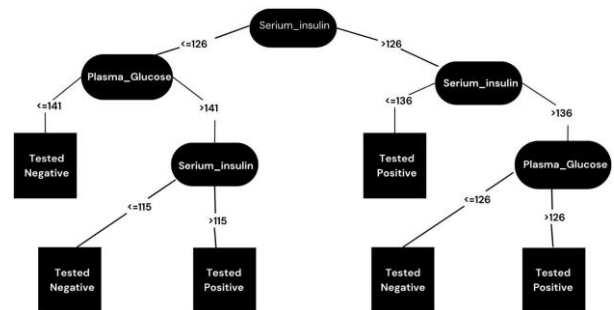


Figure 4. Utilizing a decision tree to diagnose diabetes

Data collection and preprocessing are the first steps in employing machine learning for diabetes detection. In this situation, a dataset with details on diabetes patients is required, including details on their age, gender, BMI, blood sugar levels, blood pressure, and other pertinent characteristics. In order to do this, data must be gathered from a variety of sources, including wearable technology, medical databases, and electronic health records. The information must be standardized, cleansed, and presented to make it simple to utilize for analysis. Data preparation technologies from Apache Spark include Spark SQL, Spark Data Frames, and Spark Streaming. Utilizing a decision tree to diagnose diabetes is shown in Figure 4.

4.2 Feature selection and engineering

Once the data is collected and preprocessed, the next step is to select relevant features and engineer new ones. Feature selection involves choosing a subset of the available features that are most predictive of diabetes. By combining existing features, such as ratios, averages, and combinations, new ones are created through feature engineering. Apache Spark provides various libraries and tools for feature selection and engineering, such as MLlib, Spark ML, and Spark SQL.

4.3 Use of Hadoop with Apache Spark

The Hadoop architecture is shown in Figure 5. Large datasets are handled via the distributed computing system known as Hadoop. It is an open-source project that was created by the Apache Software Foundation. The map Reduce programming architecture, on which Hadoop is built, enables distributed processing of massive datasets across computer clusters. In this paper, we will provide an overview of Hadoop, its architecture, and its applications in the field of data analytics. The Hadoop Distributed File System (HDFS) and the Map Reduce programming style are the two fundamental parts of the Hadoop framework [12]. Data can be stored on a cluster of computers using the distributed file system known as HDFS. Large datasets can be widely processed using the Map Reduce [13] programming model as a foundation. It divides big datasets into manageable pieces and runs them concurrently over a cluster of computers. A framework for diabetes diagnosis using Apache Spark is depicted in Figure 6. Hadoop also includes several other components, including YARN (Yet another Resource Negotiator), which manages resources in a Hadoop cluster, and Hadoop Common, which provides common utilities and libraries that are used by other Hadoop components.

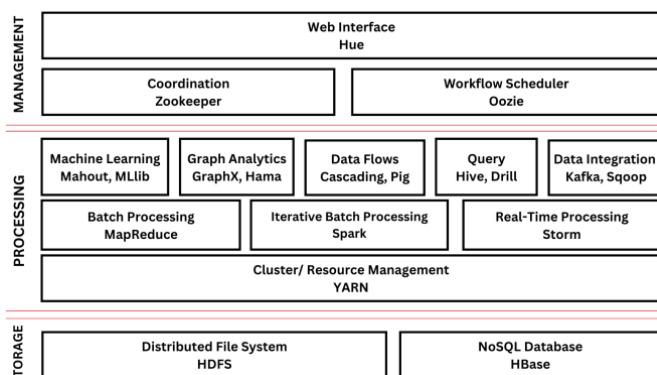


Figure 5. The Hadoop architecture

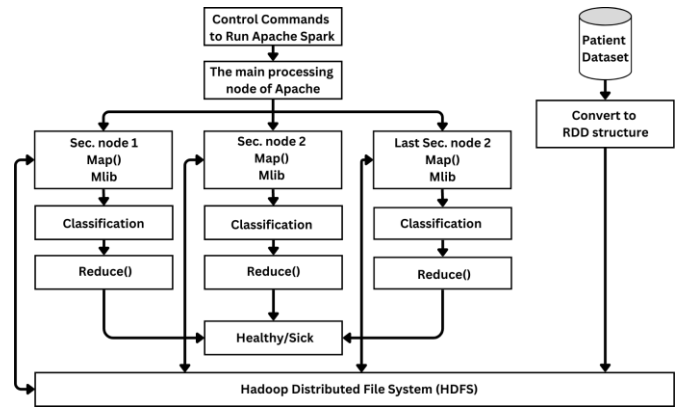


Figure 6. Framework for diabetes diagnosis using Apache Spark

4.4 Use of MapReduce to centralize decentralized data

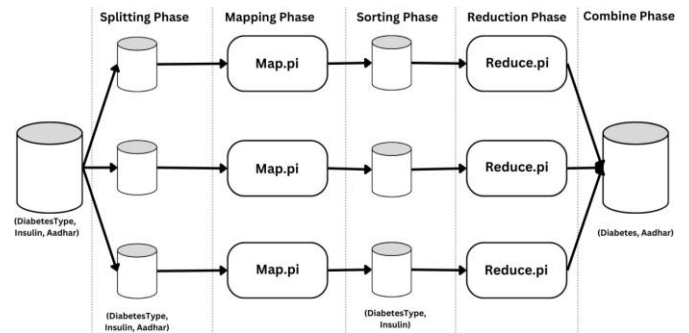


Figure 7. Steps to perform the MapReduce model

The MapReduce programming model allows processing of massive amounts of data in parallel. This model consists of a sequence of software operations, each comprising a Map stage and a Reduce step. The MapReduce operations are applied to a key, value pair and are used to process large sets of independent data [14]. These two main actions are crucial in processing vast amounts of data in a scalable and efficient manner [15]. Steps to Perform the MapReduce Model are depicted in Figure 7. The Mapping stage in the MapReduce programming model entails segmenting the input data into smaller chunks and distributing them across nodes in charge of processing. To effectively process vast volumes of data, this procedure may be repeated in a multi-level structure. Results from processing the sub-issues are passed back to the main node for additional processing. The Mapping stage in the MapReduce programming model entails segmenting the input data into smaller chunks and distributing them across nodes in charge of processing. The main node gathers the replies and outcomes obtained from the nodes during the Reduce step and carries out operations like filtering, summarizing, or converting. This technique is essential for producing the intended results from the processed data. The MapReduce paradigm divides big datasets into smaller chunks and distributes them among several nodes, allowing for efficient processing of those datasets. The MapReduce framework functions using the key-value pair approach. The Map function converts a single key-value pair into a list of ordered pairs from a single key-value pair that is supplied to it. The MapReduce framework then processes this list. One group is created out of all the pairs that share the same key. A group is made for each key generated, and the Reduce function is then

used on it. A list of values is returned by the Reduce function after receiving a key and a list of values [16]. In order to process the data, the device should have sufficient memory to hold a list of (key, values) in its main memory. The MapReduce model is efficient in processing large sets of data by dividing them into smaller segments and grouping them based on their keys. The Map and Reduce functions work together to process the data and provide the desired output.

4.5 Model selection and training, and evaluation

After data preparation and feature engineering, we can start building our machine learning model. In this case, we will use a binary classification model to predict whether a patient has diabetes or not. The next step is to select a machine learning algorithm and train it on the preprocessed data. There are various machine learning algorithms that can be used for diabetes detection, such as logistic regression, decision trees, random forests, and neural networks. Apache Spark provides various libraries and tools for model selection and training, such as MLlib, Spark ML, and Spark MLlib. We must assess the model's performance after it has been created using measures like accuracy, precision, recall, and F1 score. To improve the performance of the model, we may additionally employ strategies like cross-validation and hyperparameters tweaking.

4.6 Deployment and monitoring and results

Deploying the model in a production setting and tracking its performance over time constitute the last phase. This involves integrating the model with other systems, such as electronic health records, medical databases, and wearable devices, and ensuring that it continues to perform accurately and reliably. Apache Spark provides various tools and libraries for deployment and monitoring, such as Spark Streaming, Spark SQL, and Spark ML. To achieve accurate classification, the proposed method relies on determining the objective function of the problem precisely. Usually, the Mean Square Error (MSE) or the Root Means Square Error (RMSE) is employed as the objective function for classification. These two criteria are equivalent and are utilized to measure the quality of classification. A smaller value of MSE or RMSE indicates more precise classification. Ideally, these values tend to zero, signifying minimal classification error and maximum effectiveness of the algorithm [17]. Spark ML algorithms like decision trees, random forests, and SVM are employed in EHR analysis due to their effectiveness in handling diverse healthcare data. Decision trees offer interpretable insights, random forests enhance accuracy through ensemble learning, and SVMs are robust in predictive modeling, making them suitable for extracting valuable healthcare information from Electronic Health Records. The proposed solution leverages Apache Spark for EHR data analysis, focusing on early diabetes detection. Key innovations include integrating diverse healthcare data sources, enhancing real-time monitoring, and personalized medicine. The expected contributions are more accurate and timely diabetes diagnosis, reducing complications, and ultimately improving patient outcomes and healthcare efficiency. To effectively process vast volumes of data, this procedure may be repeated in a multi-level structure. The results are forwarded back to the main node for additional processing when the sub-issues have been resolved [18]. The use of big data analytics in the diabetic

retinopathy healthcare sector has increased recently, with Apache Spark engine as a key tool for processing and analyzing enormous amounts of healthcare data [19-23].

5. CONCLUSIONS

This study underscores the transformative potential of Apache Spark in EHR analysis and diabetes management. By harnessing this powerful framework and integrating machine learning techniques, we have demonstrated the efficient and scalable processing of EHR data, promising substantial enhancements in healthcare outcomes. The major contributions of this work lie in streamlining EHR analysis, enabling real-time insights, and actionable decision-making for healthcare professionals. Apache Spark's distributed processing and in-memory capabilities reduce runtime and enhance scalability, making it a versatile tool for healthcare data analysis. Nevertheless, this study has limitations. Future work should address challenges of interoperability, ethics, and data privacy, and explore the integration of diverse healthcare data sources. Additionally, enhancing model interpretability and deploying these solutions in clinical practice are critical for real-world healthcare applications. This study advances the healthcare sector by showcasing the immense potential of Apache Spark and machine learning in EHR analysis, promising better patient care and cost reduction.

REFERENCES

- [1] Zeadally, S., Wu, L. (2018). Certificateless public auditing scheme for cloud-assisted wireless body area networks. *IEEE Systems Journal*, 12(1): 64–73. <https://doi.org/10.1109/JSYST.2015.2428620>
- [2] Yang, S., Huang, Z., He, J., Wang, X. (2018). Type 2 diabetes mellitus prediction model based on data mining. *Informatics in Medicine Unlocked*, 10: 100–107. <https://doi.org/10.1016/j.imu.2017.12.006>
- [3] Alnazzawi, N., Al-Nemrat, A., Zeki, A.M., Hassanien, A.E. (2020). Clustering electronic health records using Apache Spark. *BMC Medical Informatics and Decision Making*, 20(1): 1-13.
- [4] Nazari, E., Shahriari, M.H., Tabesh, H. (2019). BigData analysis in healthcare: apache hadoop, apache spark and apache flink. *Frontiers in Health Informatics*, 8(1): 14. <https://doi.org/10.30699/fhi.v8i1.180>
- [5] Le, Q.V., Nguyen, H.T., Tran, T., Nguyen, T.T., Pham, L. (2017). Predicting diabetes complications using EHR data: A case study of a Vietnamese hospital. *Journal of Medical Systems*, 41(4): 1-9.
- [6] Li, Z., Zhang, Y., Feng, J., Yang, S., Guo, Y., Fang, J. (2019). A novel readmission prediction framework for electronic health records based on convolutional neural network and Apache Spark. *Journal of Medical Systems*, 43(2): 1-9.
- [7] Honavar, S.G. (2020). Electronic medical records - The good, the bad and the ugly. *Indian Journal of Ophthalmology*, 68(3): 417-418. https://doi.org/10.4103/ijo.IJO_278_20. PMID: 32056991
- [8] Shang, Y., Jiang, K., Wang, L., Zhang, Z., Zhou, S., Liu, Y., Dong, J., Wu, H. (2021). The 30-days hospital readmission risk in diabetic patients: Predictive

- modeling with machine learning classifiers. *BMC Medical Informatics and Decision Making*, 21: 57. <https://doi.org/10.1186/s12911-021-01423-y>
- [9] Belle, A., Thiagarajan, R., Soroushmehr, S.M., Navidi, F., Beard, D.A., Najarian, K. (2015). Big data analytics in healthcare. *BioMed Research International*, 2015: 370194. <https://doi.org/10.1155/2015/370194>
- [10] Haggag, M., Tantawy, M.M., El-Soudani, M.M. (2020). Implementing a deep learning model for intrusion detection on apache spark platform. *IEEE Access*, 8: 163660-163672. <https://doi.org/10.1109/ACCESS.2020.3019931>
- [11] Shrotriya, L., Sharma, K., Parashar, D., Mishra, K., Rawat, S.S., Pagare, H. (2023). Apache Spark in healthcare: Advancing data-driven innovations and better patient care. *International Journal of Advanced Computer Science and Applications*, 14(6): 608-616.
- [12] Lee, J., Kim, B., Chung, J.M. (2019). Time estimation and resource minimization scheme for apache spark and hadoop big data systems with failures. *IEEE Access*, 7: 9658-9666. <https://doi.org/10.1109/ACCESS.2019.2891001>
- [13] Yang, H., Luan, Z., Li, W., Qian, D., Guan, G. (2012). Statistics-based workload modeling for mapreduce. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pp. 2043-2051. <https://doi.org/10.1109/IPDPSW.2012.254>
- [14] Ramírez-Gallego, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Benítez, J.M., Alonso-Betanzos, A., Herrera, F. (2017). An information theory-based feature selection framework for big data under Apache Spark. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9): 1441-1453. <https://doi.org/10.1109/TSMC.2017.2670926>
- [15] Thomas, L., Syama, R. (2014). Survey on MapReduce scheduling algorithms. *International Journal of Computer Applications*, 95(23): 9-13. <http://doi.org/10.5120/16733-6903>
- [16] Dunning, T., Friedman, E. (2015). *Real-world hadoop*. O'Reilly Media, Inc.
- [17] Saravi, F.B., Moghanian, S., Javidi, G., Sheybani, E.O. (2021). Machine learning in Apache spark environment for Diagnosis of diabetes. *Preprints 2021: 2021110200*. <https://doi.org/10.20944/preprints202111.0200.v1>
- [18] Kumar, V.N., Shindgikar, P. (2018). *Modern big data processing with hadoop: expert techniques for architecting end-to-end big data solutions to get valuable insights*. Packt Publishing Ltd.
- [19] Forkan, A.R.M., Khalil, I., Ibaida, A., Tari, Z. (2015). BDCaM: Big data for context-aware monitoring—A personalized knowledge discovery framework for assisted healthcare. *IEEE Transactions on Cloud Computing*, 5(4): 628-641. <https://doi.org/10.1109/TCC.2015.2440269>
- [20] Doyle-Lindrud, S. (2015). The evolution of the electronic health record. *Clinical Journal of Oncology Nursing*, 19(2): 153-154. <https://doi.org/10.1188/15.CJON.153-154>
- [21] Salloum, S., Dautov, R., Chen, X., Peng, P.X., Huang, J.Z. (2016). Big data analytics on Apache Spark. *International Journal of Data Science and Analytics*, 1: 145-164. <https://doi.org/10.1007/s41060-016-0027-9>
- [22] Nasir, N., Afreen, N., Patel, R., Kaur, S., Sameer, M. (2021). A transfer learning approach for diabetic retinopathy and diabetic macular edema severity grading. *Revue d'Intelligence Artificielle*, 35(6): 497-502. <https://doi.org/10.18280/ria.350608>
- [23] Handoyo, A.T., Kusuma, G.P. (2022). Severity classification of diabetic retinopathy using ensemble stacking method. *Revue d'Intelligence Artificielle*, 36(6): 881-887. <https://doi.org/10.18280/ria.360608>