International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 1, February 2023, pp. 1008~1014
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i1.pp1008-1014

Matching data detection for the integration system

Merieme El Abassi[1], Mohamed Amnai[1], Ali Choukri[1], Youssef Fakhri[1], Noreddine Gherabi[2]
[1]Laboratory of Computer Sciences Research, Faculty of Sciences, Ibn Tofail University Kenitra, Kenitra, Morocco
[2]National School of Applied Sciences of Sultan Moulay Slimane University, Khouribga, Morocco

Article Info

Article history:

Received Nov 9, 2021
Revised Sep 22, 2022
Accepted Oct 5, 2022

Keywords:

Data integration
Data matching
Data quality
Entity resolution

ABSTRACT

The purpose of data integration is to integrate the multiple sources of heterogeneous data available on the internet, such as text, image, and video. After this stage, the data becomes large. Therefore, it is necessary to analyze the data that can be used for the efficient execution of the query. However, we have problems with solving entities, so it is necessary to use different techniques to analyze and verify the data quality in order to obtain good data management. Then, when we have a single database, we call this mechanism deduplication. To solve the problems above, we propose in this article a method to calculate the similarity between the potential duplicate data. This solution is based on graphics technology to narrow the search field for similar features. Then, a composite mechanism is used to locate the most similar records in our database to improve the quality of the data to make good decisions from heterogeneous sources.

Corresponding Author:

Merieme El Abassi
Laboratory of Computer Sciences Research, Faculty of Sciences, Ibn Tofail University Kenitra
Kenitra, Morocco
Email: merieme.elabassi@uit.ac.ma

1. INTRODUCTION

Big data is like a small data but in a large amount of data with a higher complexity level, because it becomes very difficult to control it by any database management tool [1]. However, big data is characterized via a set of properties, including volume, veracity, variety, and velocity. Volume represents the size of data; it can be extended up to terabyte or more. Velocity denoted how fast the data came in. variety represents that data can be structured, semi-structured and unstructured format. Today, data has become the wealth of companies and management departments, contributing to its development. The decisions based on low-quality data can be very costly, hurting businesses, partners, and customers. Furthermore, the management departments and companies need to improve their relationships through data governance. Hence, having good data quality is very important for companies, especially when they interact with other organizations or make big decisions.

The concentrate on the structure of the data to be cleaned or integrated, in order to make some metrics and ways to solve the issues of data quality. That is what the suggested methods depend on to solve these problems. Thus, to get helpful data, we need to analyze it within the range of its usage [2]. As we know, integration projects may require some support to improve the quality of data, because there are few companies who execute the procedures of data quality management in the database or data warehouse they have created.

Currently, the problem of entity analysis is a field of research in the field of data quality [3]–[5]. Just as online mining for relationships and entities has established an extensive public knowledge base, companies, governments, and researchers can also use the true value of this data, which can only be used when multiple data sources are integrated. Entity resolution refers to the task of identifying records from the

same entity in one or more data sources [6]–[9]. When only one database is used, this strategy can be called deduplication [10]. Comparing records that might be matched in an entity collection is a secondary problem. It is impractical to use traditional methods of comparison when collecting big data. Therefore, you generally organize similar records and then compare only those records that appear in the same block to improve the efficiency of the entity resolution algorithm. However, for data across multiple data sources, there are usually different standards, so before analyzing the entity, pattern matching must be performed between the data sources. Traditional pattern matching is no longer effective for large amounts of highly heterogeneous and noisy data on the network.

Several techniques are currently used to define the likelihood of matching features. These include sorted neighborhood blocking (SN) [11], duplicate count strategy (DCS) [12], and q-gram based indexing (G-gram). These are feature resolution techniques. Basically, they operate on all records in the dataset. Then we group each record into one or more blocks, and finally we create the pairs for comparison. Therefore, these methods take a long time to identify possible pairings. The probability of error is also high because the search area is larger for large amounts of data. Consequently, to minimize these problems, we need to look for a technique that reduces search time and achieves good results.

The main focus of this study is on exploiting graph mechanisms to solve the problem of big data diversity. Here, a technique is discussed to identify matching data resulting from the integration of different types of sources such as databases, comma-separated values (CSV), spreadsheets, web services, and other formats. These duplicate data cause a lot of problems when we make decisions. For example, we might find many records representing the same record with minor differences, such as finding names in a different order, which makes us consider each record duplicate as a new one, or having some names or information errors, or using different abbreviations in each database sources. So, the goal is to convert the data into a graph where the search space for duplicate records can be reduced, the records that have shared data are identified, and then some algorithms are applied to obtain a match ratio between records that have some shared information.

The contributions of this paper are five aspects. First, the data integration problem solving is propose based on the deduplication approach. In the second aspect, the problems of data integration are described. Third, the proposal is presented, while in the fourth, the principal focus is on the evaluation of that proposed method. In the last phase, conclusions and ideas for future extensions are discussed.

2. RELATED WORK

A lot of studies have conducted many challenges of data integration which we need to overcome. Several propositions have been discussed by Yeganeh *et al.* [13], who discussed the problem of considering user preferences for data quality, within several settings and improving user satisfaction from query results. Improving query results helps the users' decision-making process and should lead to higher user satisfaction, this makes the field of data quality study interdisciplinary. Different synergies are proposed to provide comprehensive data quality solutions [14]. Several researchers have worked on grouping the dimensions of data quality into conceptual views of data, data values, and data formats. Similar to the above work, Bovee *et al.* [15] and Jarke *et al.* [16] recommended to classify the data quality dimensions based on the user's role in the data warehouse environment. Also, they propose different dimensions and sub-dimensions, based on the concept of data quality as applicability. The problem of describing the quality of data sources is at the core of data integration and exchange [17], [18]. Talburt [19] talked about entity resolution and information quality and discusses the Fellegi-Sunter theory of the relationship between records, the Stanford entity resolution framework and the algebraic model for entity resolution, which are the main theoretical models that support entity resolution. Also, the way of eliminating the redundant data and supporting the master data management programs is discussed through the concept of entity resolution and by using the Oyster open-source system [20].

In addressing the entity resolution problem, a method for distributing the workload between the different computing nodes is proposed [21]. The approach is based on the use of MapReduce with standard blocking. Benny *et al.* [22] proposed similar work for entity resolution on Hadoop, which works with semi-structured data. It also contains the preprocessing tasks and the results of the comparison, indexing, and classification. The use of different classification methods and their results is described to improve future comparisons. Papadakis *et al.* [23] proposed different techniques. The first technique aims to remove the superfluous comparisons from any redundancy-based blocking method, and the second solution is used for reducing the space requirements which are mapped to the Cartesian space.

Yan *et al.* [24] developed a methodology called multi-singer that supports structured and unstructured data types, as well as tasks considering data preprocessing and comparing reduction. The objective of the work [1] is to apply deduplication techniques in different merged data sources, in order to format comparison or finding duplicate elements of the same type. Various techniques for preprocessing and testing large amounts of data are proposed [25], this system allows matching entities assigned to the same

block. It uses dynamic blocking to achieve high performance, reducing the search space and covering the same entities in blocking steps.

3. PROBLEM FORMULATION

Data inconsistency can be caused by heterogeneous data sources, which means more tools and techniques to optimize unstructured data are needed. In addition, structured data allows us to run query processes to filter, analyze, and use this data to make business decisions and build organizational capabilities [26]. Organizations face a different challenge when they need to expand to accommodate a wide range of data and create new domains. That is why a solution must be found. This involves creating high performance computing environments with advanced data storage systems. It also reduces latency while improving reliability and access to data quickly. Big data entities pose many challenges when faced with the accumulation of large data sets from different sources. Then ways of running common to the two fields to combine and execute the queries and algorithms must be found.

The wave of big data will soon spread to all areas of life, which will not only provide humans with unprecedented opportunities, but also bring major challenges. Therefore, it is very necessary to effectively solve diversity and heterogeneous data issues in big data integration [27]. Thus, the quality of information has become the latest technical requirement for users. To evaluate and enhance data quality, it is essential to implement continuous enhancement strategies. By combining data, duplicate data can be identified and then actions can be taken, like merging two similar or identical records into one. This also helps identify equally important non-duplicates, as you should know that two similar things are not the same. Overall, deduplication is a process used for moving the duplicate data in one or more databases containing information of poor quality, it is a very important technique for having good data quality. This also can be a difficult question, because the same "entity" can be referred to by different names, and these names can also contain typos, so matching ordinary strings is not enough.

For example, if you have a dataset consisting of many records, and you are trying to find records that represent the same entity, this problem is difficult to solve in a simple way. Although each entity uses a different lexical representation, direct string matching will not find duplicate records, as we can see in this example: records (1 MOBILE LIMITED, 30 CITY RD) and (1 MOBILE Ltd, 30 CITY ROAD) represent the same record entity but use a different dictionary. Therefore, in this paper, a processing technique using Sparks Rich API is proposed.

4. PROPOSED METHOD

The focus of this article is on exploiting graphs to solve the problem of big data variety. In particular, an approach is described to represent the data from multiple sources as a graph (entity, edge) showing in Figure 1. Consider the database *R={r1, r2, r3... rn}*. These registers are composed of registers with attributes *A={R.A1, R.A2, R.Am}*. Also consider the *G (ri)* function which maps the *ri∈R* record to the graph and the *S (ri, rj)* function used to measure similarity. Let us say that if *S (ra, rb)* is close to 1, record a is similar to record *b*. Typically, *G (ri)* mapping is a graphical method that can convert related data into graphs. In simple terms, an appropriate chart represents data with vertices and edges. Here, the relational data in a chart is rearranged so that the entities share common nodes.

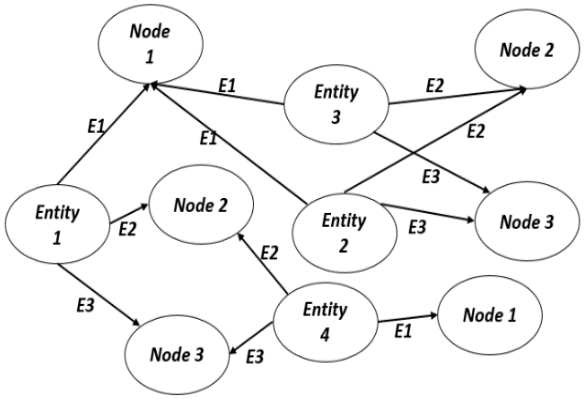

Figure 1. Schema of a graph of the dataset

Like most technologies, there are various alternative approaches to forming the essential components of a graph database. One of these methods is the property graph model, in which data is divided into nodes, relations, and attributes (data stored in the nodes or relations). Thus, nodes present instances or entities of graphs, and can contain any information called attributes. In addition, an edge represents the link between two entities, and it always needs a direction, a start and an end node, and a type. The architecture of the technology is shown in Figure 2, which is roughly divided into four steps.

a) Select the file for deduplication: The first phase summarizes the download the files sources for the deduplication technique.
b) Create graph: In the entity graph creation stage, the first step is blocking, creating two types of entities with a set of records as input, the first type belongs to the main data, and the second type belongs to other data that are matched.
c) Detect the potential duplicate's entity: The next step, perhaps the most important and also the most expensive, is to find a possible matching record. The goal is to find a record that looks like a record without having to use the same record in every field. The connection conditions are very specific: we choose to use *GraphFrameMotives*; we will remove the search space and find any duplicates. The range is wide enough to vary the amount of capture, but the selectivity is sufficient to avoid the use of cartesian products (the use of cartesian products should be avoided at all costs). Through the query of motif finding, possible pairs of duplicates are searched.
d) Compute the similarity between potential duplicates: It involves detecting the similarity between potential duplicates after computing the vector representation of the graph entities. In this approach, the term frequency-inverse document frequency (TFIDF) is chosen to be used. The weight is a statistical measure that evaluates the importance of a word to a document in a collection or corpus. Then the output variable is used to calculate the similarity between two documents represented as a vector by finding the cosine of the angle between them.

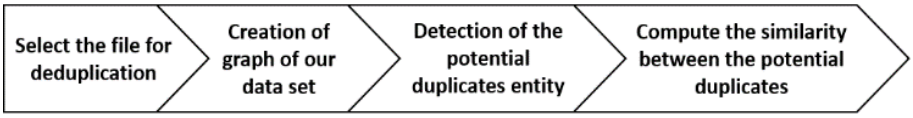

Figure 2. General architecture

5. EXPERIMENTS AND RESULTS

The method proposed in the previous section is mostly used to compute the similarity for finding the same record of duplicate elements. In this section, we evaluate it on sample input data and show the result. Table 1 indicates a partial view data set used; it keeps the information about the addresses of some companies in a file CSV.

As shown in Table 2, there is the detail of the three datasets contain the same address but it in a different format in which this type of data duplicate is not solved by the basic technique (as sorted neighborhood blocking, duplicate count strategy, or Q-Gram based indexing). All datasets contain the same address, as shown the company name is represented in a different format in each tuple of the same company. In the other colons, we have some information missing or in a different format.

Table 1. Input of sample data

ID	Company_Name	Address_Line1	Address_Line2	Post_Town	County	PostCode
1	1 MOBILE LIMITED	30 CITY ROAD	null	LONDON	null	EC1Y 2AB
2	1 TECH LTD	57 CHARTERHOUSE STREET	null	LONDON	null	EC1M 6HA
3	23 SNAPS LIMITED	16 BOWLING GREEN LANE	null	LONDON	null	EC1R 0BD
4	2E2 SERVICES LIMITED	200 200 ALDERSGATE	ALDERSGATE STREET	LONDON	null	EC1A 4HD
5	2E2 UK LIMITED	200 ALDERSGATE	ALDERSGATE STREET	LONDON	null	EC1A 4HD
6	40 50 MEDIA LTD	145-157 ST JOHN STREET	null	LONDON	null	EC1V 4PW
7	4D DATA CENTRES LIMITED	30 CITY ROAD	LONDON	Null	null	EC1 2AB
8	4GETMOBILE LIMITED	152 KEMP HOUSE	CITY ROAD	LONDON	null	EC1V 2NX

Table 2. An example of duplicate data

ID	Company_name	Address_line1	Address_line2	Post_town	Country	Postcode
1001	1 MOBILE LIMITED	30 CITY RD.	null	null	null	null
1002	1 MOBILE Ltd.	30 CITY RD	null	null	null	EC1Y 2AB
1003	1 MOBILE	null	CITY ROAD	London	null	nulls
1004	GLOBAL TECHNOLOGY	null	null	LONDON	null	null
1005	GLOBAL TECHNOLOGY Ltd	160 CITY RD	LONDON	null	null	EC1V 2NX
1006	1 TECH	57 CHARTERHOUSE	null	LONDON	null	null

As shown in Figure 3, the data set is divided into two sets. The first is called the data master contain the correct data and an identifier unique entitle *aid* is picked. The second dataset, say transaction data, that held the records, can identify the same company named *bid*. In the above example, a deterministic similarity is based on feature vectors calculated by the term frequency-inverse dense frequency (TF-IDF) method (*afeatures* and bfeatures shown in Figure 4). Then, focus on the result shown in Figure 3 to draw the chart as shown in Figure 4 of similarity between the *aid* and *bid* for detected the most similar tuples that exist in our dataset input.

Figure 4 exposed a comparison between the transaction data and the basic data when the similarity is greater than or equal to 0.4, which allows us to combine the most seven compatible records. As shown in the Figure 4, each color identifies a record in the master data and the x-axis identifies a record of transactional data, so the advantage of this approach is its ease of implementation and computational complexity, as we structure the data graphically, in any database provides us with different detection duplicates.

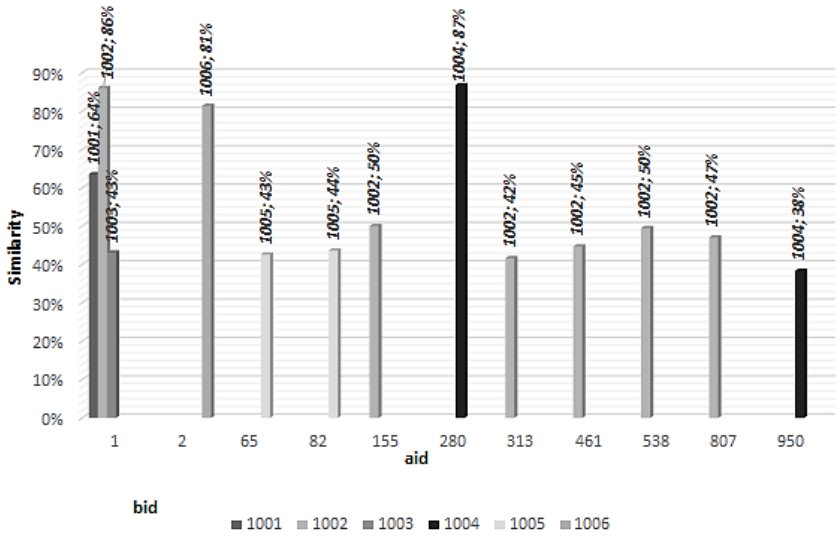

Figure 3. A sample comparison of similar

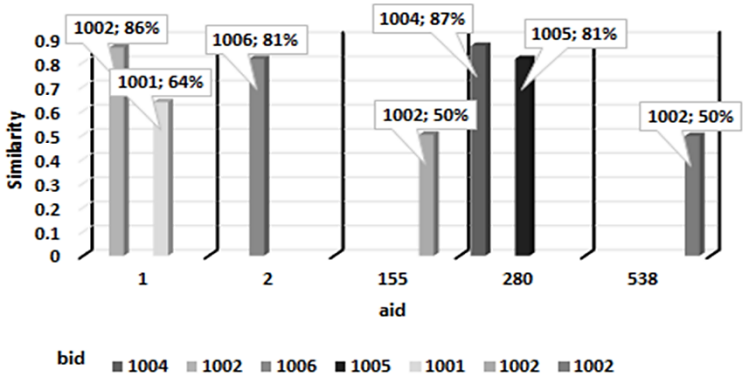

Figure 4. Comparison between the transaction data and the basic data

6. CONCLUSION AND FURTHER RESEARCH

In this paper, a method for solving the deduplication problem is presented using the Apache Spark framework and Scala language. The existing data integration is analyzed when the different formats of data combine in one format then there is a chance of duplication of data in the format as discussed. The proposed method is to compute the similarity to detect potential duplicate data. The project can be expanded to include further improvements to reduce comparisons between different registers and reduce computation time, such as using parallel computations. In future research, the method of big data integration based on Karma modeling is being explored.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewer for offering many suggestions to improve the quality of this paper. The authors acknowledge financial support from the National Scientific Research and Technology Center (CNRST) as part of the research grant program.

REFERENCES

[1] S. Garg and A. Bala, "Semantic analysis of big data by applying de-duplication techniques," *2016 International Conference on Inventive Computation Technologies (ICICT)*, Aug. 2016, doi: 10.1109/inventive.2016.7830171.
[2] A. Ben Salem, "Contextual data quality: detection and cleaning driven by data semantics," (in French), PhD Thesis, Université Sorbonne Paris Cité, 2015.
[3] I. N. Chengalur-Smith, D. P. Ballou, and H. L. Pazer, "The impact of data quality information on decision making: an exploratory analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 6, pp. 853–864, 1999, doi: 10.1109/69.824597.
[4] R. Y. Wang, H. B. Kon, and S. E. Madnick, "Data quality requirements analysis and modeling," in *Proceedings of IEEE 9th International Conference on Data Engineering*, 1993, pp. 670–677, doi: 10.1109/ICDE.1993.344012.
[5] T. Hongxun *et al.*, "Data quality assessment for on-line monitoring and measuring system of power quality based on big data and data provenance theory," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Apr. 2018, pp. 248–252, doi: 10.1109/ICCCBDA.2018.8386521.
[6] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, Jan. 2007, doi: 10.1109/TKDE.2007.250581.
[7] H. Köpcke and E. Rahm, "Frameworks for entity matching: A comparison," *Data and Knowledge Engineering*, vol. 69, no. 2, pp. 197–210, Feb. 2010, doi: 10.1016/j.datak.2009.10.003.
[8] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1537–1555, Sep. 2012, doi: 10.1109/TKDE.2011.127.
[9] X. L. Dong and D. Srivastava, "Big data integration," in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, Apr. 2013, pp. 1245–1248, doi: 10.1109/ICDE.2013.6544914.
[10] "USENIX Association," *Proceedings of FAST '11: 9th USENIX Conference on File and Storage Technologies*, 2011, Accessed: Nov. 08, 2021. [Online]. Available: https://www.usenix.org/legacy/events/fast11/tech/full_papers/fast11_proceedings.pdf
[11] L. Kolb, A. Thor, and E. Rahm, "Parallel sorted neighborhood blocking with MapReduce," *arXiv:1010.3053*, Oct. 2010.
[12] U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg, "Adaptive windows for duplicate detection," in *2012 IEEE 28th International Conference on Data Engineering*, Apr. 2012, pp. 1073–1083, doi: 10.1109/ICDE.2012.20.
[13] N. K. Yeganeh, S. Sadiq, and M. A. Sharaf, "A framework for data quality aware query systems," *Information Systems*, vol. 46, pp. 24–44, Dec. 2014, doi: 10.1016/j.is.2014.05.005.
[14] S. Sadiq, "Prologue: research and practice in data quality management," in *Handbook of Data Quality*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–11.
[15] M. Bovee, R. P. Srivastava, and B. Mak, "A conceptual framework and belief-function approach to assessing overall information quality," *International Journal of Intelligent Systems*, vol. 18, no. 1, pp. 51–74, Jan. 2003, doi: 10.1002/int.10074.
[16] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis, "Multidimensional data models and aggregation," in *Fundamentals of Data Warehouses*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 87–105.
[17] A. Doan, A. Halevy, and Z. Ives, "Describing data sources," in *Principles of Data Integration*, Elsevier, 2012, pp. 65–94.
[18] A. Doan, A. Halevy, and Z. Ives, "Incorporating uncertainty into data integration," in *Principles of Data Integration*, Elsevier, 2012, pp. 345–357.
[19] J. Talburt, *Entity resolution and information quality*. Elsevier, 2010.
[20] J. R. Talburt and Y. Zhou, "A practical guide to entity resolution with OYSTER," in *Handbook of Data Quality*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 235–270.
[21] D. Karapiperis and V. S. Verykios, "Load-balancing the distance computations in record linkage," *ACM SIGKDD Explorations Newsletter*, vol. 17, no. 1, pp. 1–7, Sep. 2015, doi: 10.1145/2830544.2830546.
[22] S. P. Benny, S. Vasavi, and P. Anupriya, "Hadoop framework for entity resolution within high velocity streams," *Procedia Computer Science*, vol. 85, pp. 550–557, 2016, doi: 10.1016/j.procs.2016.05.218.
[23] G. Papadakis, E. Ioannou, C. Niederée, T. Palpanas, and W. Nejdl, "Eliminating the redundancy in blocking-based entity resolution methods," *Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries-JCDL '11*, 2011, doi: 10.1145/1998076.1998093.
[24] C. Yan, Y. Song, J. Wang, and W. Guo, "Eliminating the redundancy in MapReduce-based entity resolution," in *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2015, pp. 1233–1236, doi: 10.1109/CCGrid.2015.24.
[25] A. C. Mon and M. M. S. Thwin, "Efficient dynamic blocking scheme for entity resolution systems," in *International Conference on Advances in Engineering and Technology (ICAET'2014)* 2014, pp. 167–171, doi: 10.15242/IIE.E0314076.
[26] A. Kadadi, R. Agrawal, C. Nyamful, and R. Atiq, "Challenges of data integration and interoperability in big data," in *2014 IEEE International Conference on Big Data (Big Data)*, Oct. 2014, pp. 38–40, doi: 10.1109/BigData.2014.7004486.
[27] W. Xiao, L. Guoqi, and L. Bin, "Research on big data integration based on Karma modeling," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017, pp. 245–248, doi: 10.1109/ICSESS.2017.8342906.

BIOGRAPHIES OF AUTHORS

Merieme El Abassi received her fundamental bachelor's degree in 2017 on Mathematics and Computer Science from Ibn Tofail University, Kenitra City, Morocco. Then, she obtained her master's degree on Big Data and Cloud Computing in 2020, from Ibn Tofail, Kenitra, Morocco. The author is currently a Ph.D. researcher in data integration. Also, she is currently doing her research in the Computer Science Research Laboratory, Kenitra, Morocco. She can be contacted at merieme.elabassi@uit.ac.ma.

Mohamed Amnai received his IEEA (Computer, Electronics, Electrical Engineering, and Automation) degree in 2000 from Errachidiacity, Molay Ismail University. In 2007, he received a master's degree from Ibn Tofail University in Kenitra. In 2011, he received his Ph.D. in Telecommunications and Computer Science from the University of Ibn Tofail in Kenitra, Morocco. Since March 2014, he has been a professor and an assistant professor at the Khouribga National School of Applied Sciences, University of Sétat, Morocco. He joined the Kénitra Faculty of Science, Department of Computer Science and Mathematics, Ibn Tofail University, Morocco, in 2018 as an Associate Professor. He is also an associate member of the Research Laboratory of the Faculty of Computer Science, Team Networks, and Telecommunications Sciences in Kenitra, Morocco. He is also an associate member of the laboratory of the IPOSI National Institute of Applied Sciences, Hassan 1 University, Khouribga, Morocco. He can be contacted at mohamed.amnai@uit.ac.ma.

Ali Choukri is an assistant professor at the National Academy of Applied Sciences. He received a master's degree in computer science and telecommunications from the University of Ibn Tofail, Kenitra, Morocco, in 2008. In 1992, he obtained a degree from ENSET (Higher Normal School of Technical Teaching). He has a Ph.D. from the School of Computer Science and Systems Analysis (ENSIAS). He works in the MIS team in the SIME laboratory, researching mobile intelligent ad hoc communication systems and wireless sensor networks. His research interests include ubiquitous computing, internet of things, delay/fault-tolerant networks, wireless networks, QoS routing, mathematical modeling and performance analysis of networks, control, and decision theory, game theory, trust and reputation management, distributed algorithms, meta-heuristics, and optimization, genetic algorithms. He can be contacted at choukriali@gmail.com.

Youssef Fakhri received a BSc (BS) in Electrophysics in 2001 and a MSc in Computing and Telecommunications (DESA) in 2003 at the Faculty of Science, Mohammed V University, Rabat, Morocco, where he obtained his M.Sc. project development in ICI company in Morocco. He received a Ph.D. 2007 by Mohammed V-University of Agdal, Rabat, Morocco, in collaboration with the Polytechnic University of Catalonia (UPC), Spain. He joined the Kénitra Faculty of Science, Department of Computer Science and Mathematics, Ibn Tofail University, Morocco, in March 2009 as an Associate Professor and as Laboratory Director of the Computer Science Research Laboratory of the Kénitra Faculty and a member of the Pole of Competencies STIC Morocco. He can be contacted at FAKHRI@uit.ac.ma.

Noreddine Gherabi Noreddine Gherabi is a professor of computer science with industrial and academic experience. He holds a doctorate degree in computer science. In 2013, he worked as a professor of computer science at Mohamed Ben Abdellah university and since 2015 has worked as a research professor at Sultan Moulay Slimane University, Morocco. Member of the International Association of Engineers. Gherabi having several contributions in information systems namely: big data, semantic web, pattern recognition and intelligent systems. He has several publications in computer science (book chapters, international journals, and conferences/workshops), and edited books. He convened and chaired more than 52 conferences and workshops. Last books in Springer: i) Intelligent Systems in Big Data, Semantic Web and Machine Learning; ii) Advances in Information, Communication and Cybersecurity; iii) Information Technology and Communication Systems. He can be contacted at gherabi@gmail.com/n.gherabi@usms.ma.