EFFICIENT STORAGE OF ETEROGENEOUS IOI DATA IN A BLOCKCHAIN USING AN INDEXING METHOD IN METRIC SPACE

K. Khettabi, B. Farou, Z. Kouahla, and H. Seridi

LabSTIC Laboratory, Department of Computer Science,
8 Mai 1945 University, Guelma 24000, Algeria

ABSTRACT

In this work, we proposed a IoT data indexing method to surpass some challenges encountered during the use of hashing in the storage of data in a blockchain. The indexing method was developed in metric space in which no dimensions are considered and only distance between objects is taken into account. The proposed method consisted on putting the index in the inner of a block. The index, called GHB-tree is based on space partitioning using hyperplane. The proposed approach was tested using two datasets of close size and different dimensions. The experimental results showed that the proposed method is efficient and competitive to other storing methods since the queries retrieve time is very reduced to be expressed by millisecond compared with that of other blockchains.

KEYWORDS

Blockchains, IoT data indexing, Metric space, k-NN search method, Retrieve time.

1. INTRODUCTION

A blockchain is a decentralized ledger that is used for the storage of transactions across a big number of computers without alteration [1]. In a blockchain, each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. In addition to the fact that a blockchain removes the possibility of tampering by a malicious actor, it presents other advantages such as cost saving and time saving. Transaction of data in a blockchain is faster because it doesn't require verification by a central authority. Nowadays, blockchain is limited to use in storing transactions for cryptocurrencies such as Bitcoin however, other uses for blockchain are developing including blockchain for monitoring of supply chains, blockchain for data sharing and blockchain for Internet of Things network management.The blockchain and IoT combination may be applied in different domains, such as IoT management [3], smart cites and smart home. However, due to the frequency of generation and timeliness of IoT data the retrieve of this data, imposes high requirements on the blockchain to store IoT data. Internet of things (IoT) systems are comprised of various devices that generate heterogeneous IoT data continuously. This continuity involved a big challenge concerning the data indexing and the query search in the dynamic IoT environment. The traditional indexing methods, such as [2]and Hollow-tree [3] became inadequate to index the big IoT data because they suffer from the issue of the degradation in large scale and they are unable to extend with the permanent collection of data. Despite the

Bibhu Dash et al: NLPAI, IOTBC -2024
pp. 125-133, 2024. IJCI – 2024 DOI:10.5121/ijci.2024.130209

existence of several indexes that aim to solve the issue of the big IoT data storage, they still inefficient. For example, the direct use of the cloud infrastructure affects negatively the communication time due to the big physical distances between the data sources and the data warehouse. In addition, with Cloud computing, data are stored in one centralized location while with blockchain technology, the store of data is done in multiple locations, with every node storing a copy of it. However, in the blockchain storage process, hashing technique presents some disadvantages such as collision risk and limited range. Indeed, Hash functions have a fixed length whatever how large the size of input data is. The hash value will always be of a fixed size which can lead to hash collisions in situations with many arriving data. In addition, hashing technique is only used in multidimensional space and this last presents some disadvantages compared with storage methods of heterogeneous IoT data developed in metric space such as BCCF-tree[4], B3CF-tree [5] and QCCF-tree [6], [7, 8]. In this work, aiming to improve the time of retrieve of heterogeneous IoT data in a blockchain, we present a new method of heterogeneous IoT data storage in blockchain based on data indexing in metric space.

The present paper is composed of five sections: introduction, related work, proposed method, experimentation, and conclusion. In the related work section, methods developed for storage of IoT data in a blockchain by indexing are described and the limitation of each method is presented. The indexing of continuous flows of heterogeneous IoT data in metric space using a GH-tree is described in the proposed method section. In the experimentation section, the computation platform and the used datasets characteristics will be presented, followed by the exposition and the analysis of the experimental results.

2. RELATED WORK

To store heterogeneous IoT data in a blockchain, some searchers has used indexing methods developed in multidimensional space. Singh et al.[9] proposed two approaches for index and query multi-dimensional historical data in blockchain. The first approach can update its index when data is generated with a high frequency. The second second approach fits well when less dynamic data is generated. Zhang et al. [10] presented a data authentication structure named GEM2- for range queries in the on-off blockchain model to minimise the overhead obtained by applying smart contracts with no sacrifice in query performance. Yao et al. [11] proposed a learned index semantic keyword query architecture to overcome the challenges of keyword queries, the data is stored as semantic information. The index is stored in the table search on the blockchain, with each block storing only the updated parts. Aslam et al.[12] introduced a decentralized RESTful storage framework which com- bines blockchain and distributed hash table (DHT) to support on-chain data editing . IoT is a very large-scale network that spans a large area. Each area contains a group of interconnected devices and generates huge amounts of continuous and heterogeneous data. This data needs to be indexed to facilitate the similarity search process.

Many indexing techniques were proposed for IoT data. In [13], a geospatial data index- ing was performed in the cloud where a parallel R-tree [14] and its parallel variants were constructed. Three construction methods were used: Apache Spark in-memory, Apache Spark on disk and MapReduce. Each one is looking for the fastest way in building, updating and executing spatial query. One of these three methods, the Apache Spark in-memory, reduces significantly the time for indexing geospatial data and querying ranges. However, this method is only used for geospatial data where the dimension is limited to three. A hierarchical multidimensional indexing method based on binary space partitioning (BSP) was proposed by Wan et al. [15] for efficient spatial query processing. After evaluating k-d-tree, quad-tree, k-means clustering and Voronoi diagram data structures, they found

that the Voronoi diagram data indexing method is suitable for general query operations with a response time of $O(log(n))$. However, the dimension limitation and the specific type of query make this method difficult to generalize. All these methods suffer from common problems. Methods developed in the cloud present numerous disadvantages such as the long distance between the end user and the cloud which, induced latency.

Our metric space method is proposed to replace hashing method when storing IoT data in a blockchain in order to minimize disadvantages presented by methods developed in the cloud. To enhance the k-NN queries retrieve in a blockchain, the proposed index is constructed by partitioning the space using hyperplane.

3. OUR PROPOSAL

The proposal approach is based on the replacement of the hatching technique, used in the storage of IoT data in a blockchain, by an indexing method using a binary tree (GH-tree) developed in metric space. After crypting and duplicating the constructed index, they are stored in blocks. in a second step, the arriving new IoT data are inserted in the existing index.

3.1.Metric Space

A metric space is defined by a distance function and a dataset. The distance function measures the similarity between two elements from the given dataset. Similar objects correspond to smaller distances. Being a metric space O, d) where O a set of points and d a distance function defined as: $d : O \times O \rightarrow R^{+}$. The distance function d is satisfying: (a) the non-negativity : $\{\forall (x, y) \in O^2, d(x, y) \geqslant 0\}$. (b) the reflexivity: $\{\forall x \in O, d(x, x) = 0\}$. (c)

the symmetry: $\{\forall (x, y) \in O^2, d(x, y) = d(y, x)\}$. (d) the triangle inequality: $\{\forall (x, y, z) \in O^3, d(x, y) + d(y, z) \leqslant d(x, z)\}$.

3.2. Space Partitionning

Space partitioning is a technique that leads to simpler data structures—and thus algo- rithms. It is based on a partitioning data, in the metric space, into two regions using two balls at a time.

For the balls construction, we choose two objects and consider them as two pivots (Figure 1). The distance between these two pivots is also the radius of the two balls. The BGH-tree nodes—or only N - is defined by:

- L leaf node a set of indexed objects: $E_L \subseteq E$ where $|E_L| \leqslant c_{max}$.
- N Internal node is a septuple: $(p_1, p_2, r, r_1, r_2, N_1, N_2) \in O^2 \times {}^3N^2$.

where:

- $r = d(p_1, p_2)$ helps to define two balls B_1 and B_2. According to Figure 1: $B_1(p_1, r)$ and $B_2(p_2, r)$, centered on p_1 and p_2 respectively and having a common radius value, large enough for the two balls to have a nonempty intersection.

- r_1 and r_2 are the distances to the farthest object in the subtree rooted at that node N with respect to p_1 and p_2, respectively.
- N_1 and N_2 are two subtrees(see Figure1,), such that: $N_1 = \{o \in N : d(p_1, o) <= d(p_2, o)\}$ and $N_2 = \{o \in N : d(p_2, o) < d(p_1, o)\}$.

3.3.GHB-Tree Construction

The GHB-tree (Figure 2) is a generalized binary tree with hyperplane for space partitioning. The construction of the GBH-tree is an incremental process. The insertion of objects is done from top to bottom. Algorithm of objects is done from the top to down. Algo- rithm 1 presents a formal description of the index construction process. Initially, the tree is empty (a leaf encompasses a set of objects). The farthest two-pivot search algorithm is used for all objects. We have considered putting in place strategies to try to balance the

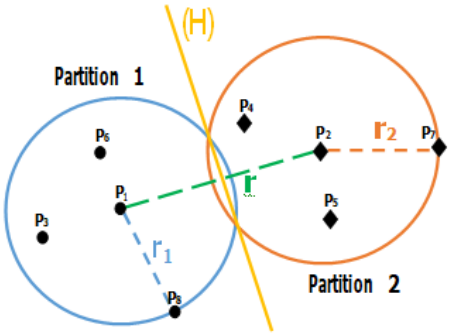

Fig. 1. Partitioning the space using hyperplane

tree, such as choosing two elements furthest apart from each other. After the container will be divided into two non-overlapping subsets so that each element of the container belongs to its nearest pivot. Then, this leaf is replaced by an internal node with p1 and p2, and two leaf nodes are created (Figure 2). Before inserting the index in a block, the root of the GHB-tree is crypted and copies of this index are created.

Algorithm 1 Construction of a GHB-tree

Construction of GHB-tree ($\in$ P()) $\in$ N
With:
(p_1, p_2)= The two farthest pivots

$$\text{GHB-tree} = \begin{cases} \perp & \text{if S=Ø} \\ (e, \perp, \perp, \perp) & \text{if S={e}} \\ p_1\ p_2 & \\ ConstructionofGHB-tree(\{e \in S : d(p_1, e) \leqslant d(p_2, e)\} \setminus \{p1\}) & \text{else} \\ ConstructionofGHB-tree(\{e \in S : d(p_2, e) < d(p_1, e)\} \setminus \{p_2\}) & \end{cases}$$

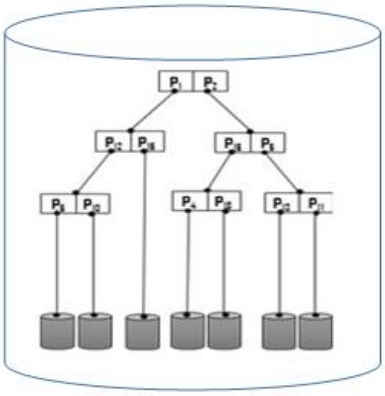

Fig. 2. GHB-tree in a block.

3.4.K-NN Retrieve

The formal description of the k-NN retrieve in the GHB-tree is summarized in Algorithm 2. The aim of the k-nearest neighbor search is to find the set A of objects closest to a query point q. The kNN search algorithm starts with a query radius r_q initialized to $+\infty$ which should lead to scanning the dataset and then decreases by traversing each tree which corresponds to the distance to the k^e object in the ordered list A. Comparing the distances d_1 and d_2 between the query point q and the two pivotsp_1 and p_2 respectively with r_q indicates the descent of the query point in the index. The leaf nodes contain a subset of the indexed data with a maximum cardinal c_{max}. To find the k nearest neighbors of a leaf, we simply sort the indexed data according to their increasing distances to the query q. As a result of the search, the first k sorted objects are returned.

Algorithm 2 Search-kNN in a bolck containing a GHB-tree

kNN-GHB-tree in a block $N \in \mathbb{N}$, $q \in \mathbb{R}^n$, $k \in \mathbb{N}^*$, $d : O \times O \to \mathbb{R}^+$, $r_q \in \mathbb{R}^+ = +\infty$, $A \in (\mathbb{R}^+ \times O)^N = \emptyset$

with :

- $A = ((d_1, o_1), (d_2, o_2), \ldots, (d_{k'}, o_{k'}))$;
- $d_1 = d(p_1, q)$;
- $d_2 = d(p_2, q)$;
- $C_1 = B(q, r_q) \cap B(p_1, r) \neq \emptyset$, for the intersection ;
- $C_2 = B(q, r_q) \cap B(p_1, r) \neq \emptyset \wedge B(q, r_q) \cap B(p_2, r) \neq \emptyset$, for the partial ball centered on p_1 ;
- $C_3 = B(q, r_q) \cap B(p_1, r) \neq \emptyset \wedge B(q, r_q) \cap B(p_2, r) \neq \emptyset$, for the partial ball centered on p_2 ;
  - $A_0 = A$;
  - $C_0 = \text{true}$;
  - $r_{q_0} = \min\{r_q, d_{k'}\}$ if $k' = k$ else r_q ;
  - $A_i = \text{kNN-GHB-tree in a block}(N_i, q, k, r_{q_{i-1}}, A_{i-1})$ if C_i else A_{i-1} ;
  - $r_{q_i} = \min\{r_{q_{i-1}}, d_k\}$ if $|A_{i-1}| = k \wedge A_{i-1} = ((d_1, o_1), \ldots, (d_k, o_k))$ else $r_{q_{i-1}}$.

$\triangleq$ $A, k\text{-sort}(A \cup \{(d(o, q), o) : o \in L\})$ if $N = L$
A_2 if $N = (p_1, p_2, r, N_1, N_2)$

4. Experiment and Results

To test and compare the efficiency of the proposed approach, experiments were performed on two real data sets with different sizes and dimensions. These two databases have been carefully selected from among others to bring together most of the problems encountered in storing

IoT data in a blockchain and are presented as follow:

1. GPS trajectory: a dataset of 16000 3D vectors, containing transport trajectories in the northeast of Brazil [5].
2. Tracking of a moving object: a real dataset of 15000 20D vectors, representing the results of a random simulation of tracking a moving object using wireless cameras [4]

The experiments were performed using the Python programming language installed on an Intel®CoreTM i7-8550UCPU, 1.80 GHz*8 processor with a 64-bit Linux operating system (Ubuntu). The experiments were performed by simulating arrival new data of size 4000 for GPS trajectory dataset and 5000 for tracking of moving objects dataset.

4.1.Evaluation of K-NN Retrieve Results

Figure 3 presents the variation of the k-NN retrieve time in each block as a function of the k parameter for GPS trajectory dataset. We remind that this dataset is composed of objects of dimension 3. One can see that the time of retrieve for all values of k is clearly reduced to be expressed by millisecond. For the same parameter k, no significant variation in the time of k-NN retrieve is observed in spite the increase of this parameter. for example, for k=5, the time of retrieve of queries in the fist block of 4000 objects is 0.10014 ms while in the fourth block of 16000 objects, the time of queries retrieve is 0.10085 ms. We can also that the time of retrieve varied as a function of the k parameter. In the last block, the time of search increases to attend 0.10133 ms for k=20 which represents a variation of 0.05 percent. The variation, as a function of the k parameter, of the k-NN retrieve time in each

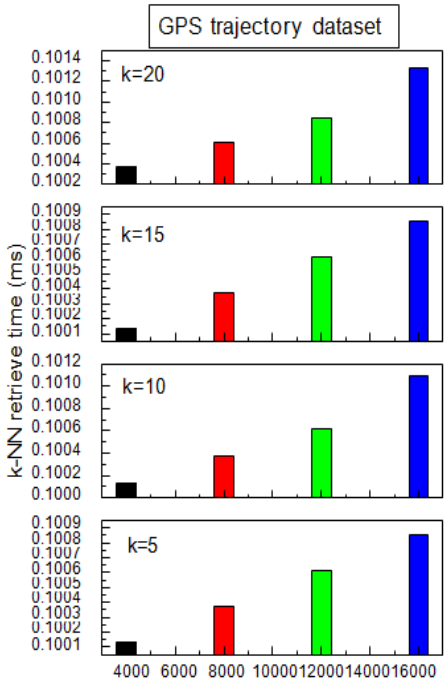

Fig. 3. Variation of the k-NN retrieve time in a blockchain containing GPS trajectory dataset of 3D objects

block is presented in figure 4 for the tracking dataset which contains objects of dimension 20. We can see that the retrieve time is also expressed by millisecond and this traduces the efficiency. The same remarks could done for the retrieve results of this dataset. In the last block, as a function of the k parameter, the retrieve time increases from 1.34 ms for k=5 to reach 1.48 ms for k=20 with a variation of about 0.1 percent. For the same k parameter, the retrieve time varies slightly except for k=10 where the retrieve time remains invariant. The time of k-NN retrieve in blockchain containing GPS trajectory objects of dimension 3 is less than that in blockchain containing tracking dataset objects of dimension 20 by 93 percent and this indicates that the proposed approach is not sensitive to the objects dimension. The retrieve time results using k-NN method in a blockchain containing GHB- trees developed in metric space evidenced the efficiency of the proposed approach when storing heterogeneous IoT data in a blockchain. According to Chen et al.,[16] for retrieve point object in Verkle AR*-tree the time of search is 1s in block size =160. For tracking

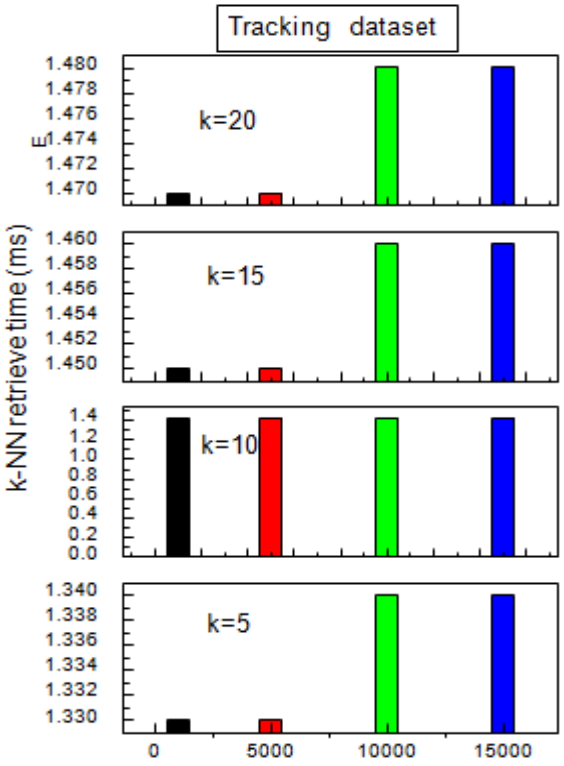

Fig. 4. Variation of the k-NN retrieve time in a blockchain containing tracking dataset of 20D objects

data set the time of search is 1.34 ms for k=5, this indicates that the use of our proposed approach improves the efficiency of retrieve data in blockchain.

5. CONCLUSION

In this paper, we proposed a method based on data indexing to avoid some challenges encountered during the use of hashing in the storage of data in a blockchain. The indexing method was developed in metric space in which no dimensions are considered and only distance between objects is taken into account. The proposed index, called GHB-tree is based on space partitioning using hyperplane. The proposed approach was tested using two datasets of close size and different dimensions. The experimental results showed that the proposed

method is efficient and competitive to other storing methods since the queries retrieve time is very reduced compared with that of other blockchains. AS future work, we will append a cryptographic function to each root of the tree and use a distributed system to simulate plockchain technology.

REFERENCES

[1] U.Majeed,L.U.Khan, I.Yaqoob,S.M.A.azmi, K.Salah, C.s.Hong, Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges, J. Netw. Comput. Appl, 2021, 181, 103007

[2] V.Srinivasan, and M.J.Carey, Performance of b-tree concurrency control algorithms, in Proceedings of the 1991 ACM SIGMOD international conference on Management of data,1991, pp. 416–425.

[3] S.Brinis, C.Traina, and Traina, A. J. Hollow-tree: a metric access method for data with missing values. Journal of Intelligent Information Systems,2019, vol. 53(3), pp. 481–508.

[4] A.E.Benrazek, Z.Kouahla, B.Farou, M.A. Ferrag, H.Seridi, and Kurulay, M. An efficient indexing for internet of things massive data based on cloud-fog computing. Transactions on emerging telecommuni- cations technologies, 2020, vol. 31(3), pp. e3868.

[5] k.Khettabi, Z.Kouahla,, B.Farou, H.Seridi and M.A.Ferrag, Clustering and parallel indexing of big iot data in the fog-cloud computing level. Transactions on Emerging Telecommunications Technologies,2022, vol. 33(7), pp.e4484.

[6] K.Khettabi,Z.Kouahla,, B.Farou, and H.Seridi, QCCF-tree: A new efficient iot big data indexing method at the fog-cloud computing level. In 2021 IEEE International Smart Cities Conference (ISC2),2021, pp. 1–7. IEEE.

[7] K.Khettabi, Z.Kouahla, B.Farou, H.Seridi, and M.A.Ferrag, A new method for indexing continuous iot data flows in metric space. Internet Technology Letters,2023, vol. 6(6),pp. e391.

[8] K.Khettabi, Z.Kouahla, B.Farou, H.Seridi, and M.A.Ferrag, Efficient method for continuous iot data stream indexing in the fog-cloud computing level. Big Data and Cognitive Computing, 2023, vol. 7(2),pp. 119.

[9] B.C.Singh, Q.Ye,H.Hu, and B.Xiao, Efficient and lightweight indexing approach for multi-dimensional historical data in blockchain, Future Generation Computer Systems,2023, 139, pp.210–223.

[10] C.Zhang, C.Xu, J.Xu, Y.Tang, B.Choi, GEM(2)-Tree: A Gas-Efficient Structure for Authenticated Range Queries in Blockchain, In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE 2019), Macao, China, 8–11 April 2019, pp. 842–853.

[11] Z.Yao, J.Xin, K.Hao , Z.Wang, W.Zhu, Learned-Index-Based Semantic Keyword Query on Blockchain. Mathematics,2023, vol.11(9), pp.2055.

[12] S.Aslam, M.Mrissa, A RESTful Privacy-Aware and Mutable Decentralized Ledger. In Proceedings of the 25th European Conference on Advances in Databases and Information Systems (ADBIS), Univ Tartu, Inst Comp Sci, Tartu, Estonia,2021, pp. 193–204.

[13] S.V.Limkar, and R.K.Jha, A novel method for parallel indexing of real time geospatial big data generated by iot devices. Future generation computer systems,2019, vol. 97, pp. 433–452.

[14] A.Guttman, R-trees, A dynamic index structure for spatial searching. In Proceedings of the 1984 ACM SIGMOD international conference on Management of data,1984, pp. 47–57.

[15] S.Wan,, Y.Zhao, T.Wang, Z.Gu, Q.H.Abbasi, and K.K.R.Choo, Multidimensional data indexing and range query processing via voronoi diagram for internet of things, Future Generation Computer Systems,2019, vol. 91, pp. 382–391.

[16] H.Chen , D.Liang , Adaptive Spatio-Temporal Query Strategies in Blockchain, ISPRS International Journal of Geo-Information, 2022, vol. 11(7), pp. 409.

AUTHORS

Karima Khettabi received PhD in Computer Science from the University of 08 mai 1945-Guelma (Algeria) in 2023. She is a Masters graduate from the University of Skikda (Algeria). Her Masters focused on the use of machine learning for spams detection. Cur- rently, her research interest is developing efficient methods for indexing big IoT data in metric space.

Brahim FAROU Received State Engineer degree in computer science systems from Na- tional School of Computer Science (Algiers, Algeria) in 2006 and the Magister degree in the sciences and technologies of information and communication from Guelma University in 2009. He received his DSc in Computer Science with distinction in 2016 and the HDR degree with distinction in 2018 from the University of Annaba, Algeria. He is currently associate professor in the computer science department, Guelma University and GADM team member at LabSTIC laboratory. He also occupied many administrative positions, he was deputy head of department responsible for teaching from 2010 to 2012 and he is currently deputy head of the department responsible for post-graduation. His research interests include color constancy, video mining, object recognition, object tracking, IoT, surveillance systems and computer vision.

Zineddine KOUAHLA Lecturer-researcher at the University of Guelma 08 May 1945. Graduated from Burgundy university in 2008, I defended my doctoral thesis in February 2013 at the University of Nantes. The thesis, with Indexation in the metric spaces Index tree and parallelization. My current research areas are:Multimedia databases: multimedia information modeling and structuring, content search, hypermedia navigation Implement systems for recognizing, indexing, searching and classifying multimedia documents.

Hamid SERIDI Received his Bachelor's degree with honours in 1981, from the University of Annaba, Algeria, and the Master's degree from the Polytechnic Institute of New-York, USA in 1984, both in Electrical Engineering. He received his PhD in Computer Science with distinction in 2001 from the University of Reims, Champagne Ardenne, France. He was Vice Dean of the Post-Graduation, Scientific Research and External Relations in the University of Guelma. Currently he is Professor and Director of Laboratory of Sci- ence and Information Technologies and Communication "LabSTIC," http://labstic.univ-guelma.dz/fr. He is also Chairman of the Scientific Council of the Faculty of Mathematics and Computing and Material Sciences. He is an expert member at the national committee for evaluation and accreditation national projects research. His research interests include approximate knowledge management, pattern recognition and artificial intelligence, data mining, video mining, machine learning, and cryptography.