A survey on Indexing Techniques for Big Data: Taxonomy and Performance Evaluation

4 authors, including:

Abdullah Gani
University of Malaya
212 PUBLICATIONS   **5,315** CITATIONS

SEE PROFILE

Aisha Siddiqa
University of Malaya
9 PUBLICATIONS   **248** CITATIONS

SEE PROFILE

Fariza Hanum Nasaruddin
University of Malaya
31 PUBLICATIONS   **310** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Improving QoS Of WiMAX By On_Demand Bandwidth Allocation Based On PMP Mode. View project

GAUChO: A Green Adaptive Fog Computing and Networking Architecture View project

SURVEY PAPER

A survey on indexing techniques for big data: taxonomy and performance evaluation

Abdullah Gani[1] · **Aisha Siddiqa**[1] · **Shahaboddin Shamshirband**[1] · **Fariza Hanum**[1]

Abstract The explosive growth in volume, velocity, and diversity of data produced by mobile devices and cloud applications has contributed to the abundance of data or 'big data.' Available solutions for efficient data storage and management cannot fulfill the needs of such heterogeneous data where the amount of data is continuously increasing. For efficient retrieval and management, existing indexing solutions become inefficient with the rapidly growing index size and seek time and an optimized index scheme is required for big data. Regarding real-world applications, the indexing issue with big data in cloud computing is widespread in healthcare, enterprises, scientific experiments, and social networks. To date, diverse soft computing, machine learning, and other techniques in terms of artificial intelligence have been utilized to satisfy the indexing requirements, yet in the literature, there is no reported state-of-the-art survey investigating the performance and consequences of techniques for solving indexing in big data issues as they enter cloud computing. The objective of this paper is to investigate and examine the existing indexing techniques for big data. Taxonomy of indexing techniques is developed to provide insight to enable researchers understand and select a technique as a basis to design an indexing mechanism with reduced time and space consumption for BD-MCC. In this study, 48 indexing techniques have been studied and compared based on 60 articles related to the topic. The indexing techniques' performance is analyzed based on their characteristics and big data indexing requirements. The main contribution of this study is taxonomy of categorized indexing techniques based on their method. The categories are non-artificial intelligence, artificial intelligence, and collaborative artificial intelligence indexing methods. In addition, the significance of different procedures and performance is analyzed, besides limitations of each technique. In conclusion, several key future research topics with potential to accelerate the progress and deployment of artificial intelligence-based cooperative indexing in BD-MCC are elaborated on.

✉ Shahaboddin Shamshirband
shamshirband@um.edu.my

1 Department of Computer System and Information Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

Ⓐ Springer

1 Introduction

As information technology alters our lifestyle, the collection of digital information in terms of structured and non-structured [1] data known as big data is rapidly developing. Big data is definitely a phenomenon with direct impact on quality of life. Applications of big data can be found in mobile cloud computer systems, such as divisions of purchase transactions [2], social networks [3], teacher commentary [4], e-science [5], and healthcare systems [6]. However, efficiently handling such amounts of data is a challenge. In order to analyze big data on cloud, efficient processing of indexing techniques should be designed [7]. Existing technologies do not even come close to meet the indexing requirements of big data; they are not fully formed to analyze such distributed, clustered, and multi-dimensional terabyte to petabyte scale data [8].

Indexing is applied in big data to perform retrieval tasks from voluminous, complex datasets with scalable, distributed storage in cloud computing [9]. It is impractical to perform manual exploration on such records and an efficient, high-throughput indexing technique would optimize the performance of data query operations [10]. Therefore, efficient indexing techniques are required to effectively access big data. Researchers have used various indexing procedures with focus on big data. For example, semantic indexing-based approach facilitates enhanced and precise searches for big data in cloud [11], a file index for efficient event stream indexing in terms of large text collection in cloud computing [12,13] and R-tree-based indexing to provide multi-dimensional data indexing in cloud [7]. Indexing with a reinforcement agent adjusts to make a trade-off between power and performance [14]. As big data leads to a complex clinical field, less cost and time [15] are required to analyze data with indexing techniques; thus, data indexing efficiently contributes to reduce time although high costs must still be tolerated while developing such techniques. According to big data requirements (i.e., volume, velocity, variety, veracity, value, variability, complexity), an efficient indexing technique applied to big data needs to satisfy these requirements [9].

There are many indexing techniques available in the literature. However, there is no state-of-the-art survey investigating the performance and consequences of techniques for solving indexing in big data issues. There is no guidance to help researchers in comparing or selecting indexing techniques. Hence, a guidance/comparative study is necessary. In this research, the developed taxonomy is intended to help researchers understand the performance of different indexing techniques from the perspective of big data indexing requirements. In taxonomy, indexing techniques are categorized based on three methods which are non-artificial intelligent (NAI), artificial intelligent (AI), and collaborative artificial intelligent (CAI) techniques.

Traditional or non-artificial intelligence indexing approaches, such as bitmap indexes [16], graph query processing [17], and tree-based indexing, perform reasonably regarding volume, velocity, variety, variability, value, and complexity, but they fail to detect 'unknown' behavior or unknown big data. Therefore, intelligent indexing in terms of a single classifier is a substitute to monitor requirements of big data with continuously changing behavior, but it may exhibit more time in query result generation compared to traditional indexing, something considered inefficient. A method based on NAI entails means such as bitmap, hashing, B-tree, and R-tree, which operate as classifiers for indexing. A rule-based automatic indexing implements cover-known patterns to improve text categorization performance; thus, for unknown pattern changes, it is not feasible [18]. To overcome this issue with unknown

changes, artificial intelligence (AI) has proved to be a more accurate detection method in constructing a hybrid indexing mechanism capable of automatically detecting inconsistent activities [19]. A case-based data clustering method is integrated with fuzzy decision tree (FDT) to develop a hybrid model that produces efficient results for test data [20]. Indexing techniques developed on the basis of artificial or non-artificial intelligence methods may not exactly satisfy big data indexing requirements, but the promotion of AI as fuzzy as well as machine learning and manifold learning is widely agreed on. With the collaboration of AI or NAI-based indexing methods, previously mentioned problems may not be faced. Consequently, three categories of indexing techniques, traditional or NAI, artificial intelligence (AI), and collaborative artificial intelligence (CAI), are highlighted in this study.

From a research perspective, different indexing techniques have been reviewed based on the NAI, AI, and CAI methods to map with the challenges of big data. Thus, the survey leads to the development of taxonomy of indexing techniques for big data in mobile cloud computing. It intends to provide a study of state-of-the-art indexing techniques in the field of big data on cloud and addresses major issues in this regard. Consequently, the intent of this paper is to address the following:

- To identify and evaluate big data indexing techniques in cloud computing.
- To classify indexing techniques used in big data.
- To design taxonomy and analyze the indexing techniques based on big data indexing requirements.

The remainder of this research paper is organized as follows. Section 2 presents indexing techniques recently proposed by researchers along with datasets on which these techniques are implemented. Indexing requirements for big data are additionally described. Section 3 states the approach applied to categorize these techniques, and taxonomy of indexing methods is presented in Sect. 4. Analysis and mapping of existing techniques in each category with big data indexing requirements are addressed in Sect. 5, while the viability of AI and CAI techniques is provided in Sect. 6. Section 7 concludes the debate by providing the significance of this study and suggesting new future directions.

2 Big data indexing techniques

The latest research findings on indexing techniques suggest ways of improving indexing accuracy performance so the indexing quality does not deteriorate. These techniques serve to optimize search performance in big data with a better time-space index trade-off. This section elaborates big data indexing requirements that make it prominent from conventional data. According to these requirements, every indexing technique will be analyzed for its feasibility and applicability to big data in mobile cloud computing. Existing indexing techniques are discussed, and datasets utilized to test their performance are also provided in this section.

2.1 Big data indexing requirements

Accuracy and timeliness are the basic measures presenting the effectiveness of indexing methods. Identifying the exact characteristics of an indexing technique is essential to establish big data requirements and develop a system. In line with the nature of big data to compare indexing techniques, seven possible big data challenges are considered as requirements, namely volume, velocity, variety, veracity, variability, value, and complexity (6Vs and Complexity). As these characteristics are the generally concerned matters to describe big data, Chang

et al. [21] have also considered most of them to elaborate big data as new paradigm. They affirmed that 'volume' is not the only one distinguishable feature for its 'bigness,' but variety, velocity, and value are also significant in this contribution. Furthermore, other characteristics like veracity, variability, and complexity are also discussed regarding big data. Furthermore, Hashem et al. [22] have adopted these characteristics to propose a definition of big data. They explained volume, variety, velocity, and value to describe large amount of miscellaneous form of scalable data. They stated that big data refers to such techniques and technologies which are applied on large and complex data to derive value from it. In our study, we describe these features of big data and contemplate them to analyze the propriety of discussed indexing techniques for big data. The 6Vs and complexity requirements and challenges are described as below:

- *Volume* Managing and indexing extensive volumes is the most evident challenge with big data [9]. The word 'big' in big data defines the volume in itself. At present, existing data is in petabytes and is expected to increase to zettabytes in the near future [23].
- *Velocity* As a challenge, velocity comes with the need to handle the speed at which new data is created or existing data is updated [9]. Nowadays, e-commerce has not only had great impact on the speed of data, but the richness of data has also increased, which is exploited in different business transactions, e.g., Web site clicks [24].
- *Variety* Big data implementation requires handling data from various sources, such as Web pages, Web log files, social media sites, e-mails, documents, and sensor device data. Clearly, big data can have different formats and models. This brings forth the challenge of big data variety [24–26].
- *Veracity* To what extent can data be trusted when key decisions need to be made regarding big data collected at high rates? Simply knowing that data is in fact sufficiently accurate, not spoofed, not corrupted, or comes from an expected source is difficult. This is an important issue known as big data veracity [27].
- *Variability* It deals with inconsistencies in big data flow. Data loads become hard to maintain, especially with the increasing usage of social media that generally causes peaks in data loads when certain events occur [23].
- *Value* How do large amounts of data influence the value of insight, benefits, and business processes? The challenge of big data value is directed at data usefulness in decision-making. Any user can execute certain queries for stored data and figure out important results from the filtered data obtained. It has been noted that 'the purpose of cloud computing is insight, not numbers' [24,28].
- *Complexity* This feature deals with the degree of interconnectedness, possibly very large, and interdependencies in big data structures [24]. Nowadays, as data has more than one source, its linking, matching, cleansing, and transformation across systems are significant activities. However, connecting and correlating relationships, hierarchies, and multiple data linkages are also very important. If complexity in terms of these objectives is not considered, big data cannot be organized better [29].

The above-mentioned challenges for big data indexing are used to evaluate indexing techniques. On the subject of volume, an indexing method is tested to observe that for how much data volume, it maintains performance. Discussed techniques are analyzed for different data formations and updating speeds to observe their behavior. Performance is analyzed with respect to various data types and formats. Furthermore, accuracy, inconsistency, significance to decision-making, and degree of interconnectedness are additional measures for evaluation.

2.2 Current indexing techniques

State-of-the-art indexing techniques utilize large datasets with a variety of data types. They are designed to cope with certain data requirements; thus, they are suitable for specific situations. A hash-based indexing technique presented by Zhu et al. [30] is designed to be search-efficient in the context of high-dimensional data. The compact Steiner tree (CST), an extension of the Steiner tree, is developed to allow for improved keyword searches on relational databases [31]. This study discusses techniques that are currently available for data indexing and examines their applicability in data domains besides their suitability in some conditions like search cost, resource consumption, result accuracy, and type of queries to be applied.

B-tree-oriented indexing corresponds to an audit of multi-dimensional big data pertaining to a set of rules and operators, which provide competent mapping of related data with search keys. It is capable of dealing with records of different lengths that are commonly observed in big data [32]. B-tree-based indexing method draws a collection of p-samples for a series of geometrically decreasing values of temporal data [33]. It takes near-linear time to answer any top-k (t) query with optimal I/O cost expected but is impractical when dealing with unknown behavior of online data streams. In addition, the method faces high query cost depending on the internal structure of indexing. To alleviate the drawbacks of online indexing and time complexity, a hybrid indexing classifier that considers dynamic graph partitioning is recommended for a later stage to augment detection rate by dynamically adjusting the graph operator. The most remarkable advantages of the hybrid B-tree for trajectory data indexing are robustness and flexibility. To moderate the veracity correlation, Sandu Popa et al. [34] have suggested hybrid B-tree indexing that combines graph partitioning and a set of composite B+-tree local indexes. The B-tree algorithm performs robust indexing but consumes vast computing resources when carrying out online data stream indexing.

Another tree-based indexing method introduces a rule to divide and merge nodes to enhance multi-attribute access performance [35]. A novel key design based on an R+-tree (KR+-index) is developed by Wei et al. [36] for efficiently retrieving skewed spatial data. Their method takes CDM (cloud data management) characteristics into account, and the KR+-index is proved to be more efficient than other indexing techniques. Experiments show that the KR+ technique performs better than state-of-the-art methods, especially for skewed data. This technique is efficient for data accessing and provides support not only with range queries but also with nearest-neighbor queries.

For relational databases, it is common to apply keyword searches to reduce the Steiner tree costs in database graphs [31]. The objective of the CS tree is to reduce the implementation cost of Steiner trees for performing keyword searches on relational database management systems. An effective structure-aware index that appears in the form of relational tables can be seamlessly integrated into any existing relational database management system (RDBMS). RDBMS capabilities can be explored to effectively and progressively identify the top-k relevant CS trees using SQL statements. It takes less average elapsed time and assures greater accuracy compared to other approaches.

Bitmap indexes are known as the most effective indexing methods for range queries on append-only data [16]. A bitmap indexing technique is based on bulk index data stored as sequences of bits. The bit sequences are utilized in bitwise logical operations to answer queries. Bitmap indexes are classified into two groups: multi-level and multi-component compressed bitmap indexes. A bit-sliced index is constructed from binary encoding schemes by decomposing the bin numbers (a set of identifiers) into multiple components [37]. Although

a bit-sliced index utilizes binary encoding to produce the minimum number of bitmaps, it fails to respond to queries in a minimum time.

Hash-based methods perform faster approximate similarity searches for high-dimensional data. They are applied to many kinds of real applications, such as image retrieval, document analysis, and near-duplicate detection [38]. Sparse hashing (SH) performs fast approximate similarity searches to generate interpretable binary codes. It converts the original feature space of data into low-dimensional space by proposing a nonnegative sparse coding method [30]. The main objective of hashing is to represent high-dimensional data with compact binary codes to obtain faster search results.

HubRank [39] is a system developed on the basis of Berkhin's Bookmark Coloring Algorithm (BCA) which provides a graph database and an efficient index to respond PageRank queries consistently. For experimental analysis, entity relationship (ER) graph are collected from CiteSeer and US patents. Real queries from CiteSeer are also considered, and the results show the efficiency of HubRank in index size, index construction time, and high accuracy with query processing time.

In semantic-based indexing, each annotation of every document is stored in a database and a weight is assigned to reflect how relevant the ontological entity is to the document meaning. The main idea is that the more semantically closed concepts appear in a document, the higher the vector values these concept dimensions obtain [40]. Semantic-based indexing can be a basis for enhanced search processes for big data in cloud. A semantically enhanced cloud service environment is developed through ontology evolution to facilitate the discovery of cloud resources meeting users' needs [11].

A state SVM network is developed by Chen-Yu et al. [41] for modeling human behavior in surveillance videos. Paul et al. [14] has followed Chen-Yu's basic architecture, but several changes to support the calculation of state and transition probability are included. Similar to hidden Markov model (HMM), the state SVM network also consists of several states, each of which is modeled by a two-class SVM. A state SVM is able to generate a probabilistic score according to the input frame. Alternatively, the connection among states in the state SVM network is represented by an edge, and the transition probability is computed from training data.

In content-based multimedia retrieval, manifold learning is usually applied to the low-level descriptors of multimedia content in order to facilitate nonlinear dimensionality reduction [42–44]. This concept can be easily extended to address cross-modal or multi-modal retrieval problems, where descriptors from different modalities are mapped to the same low-dimensional manifold and reduce the multi-modal similarity matching problem to a simple distance metric on this manifold. Majority of manifold learning approaches are based on computing the k-nearest neighbors among all dataset items in order to construct an adjacency matrix.

Fuzzy rule base is beneficial in indexing instances but is impractical when dealing with unknown events in big data cases. To lessen the drawbacks of unknown big data indexing, hybrid fuzzy classifiers that consider dynamic fuzzy rule tuning are recommended for a later stage to augment detection rate by dynamically adjusting the rules [45]. Fuzzy indexing works by encompassing a large number of moving objects and takes sub-seconds for index creation [46]. It captures images at a maximum possible frame rate to better use them for indexing. A given index image serves to answer incoming queries in a very short span of time. Following exploitation, the index image is discarded to free up space for a new index image, making the index memory efficient and up-to-date. The main advantages of fuzzy indexing are that it supports predictive queries, and it has high query throughput, low query response time, and high update performance.

Multiple users have their own internal knowledge representations (concepts and mental categories), and they interact with social tagging systems by assigning tags to numerous documents as they utilize information through the system. Internal knowledge representations partially reflect the users' background knowledge as well as differences in their information needs. The correlations between users, tags, and indexed documents define the external system folksonomies from where users can explore and learn. A computational social learning model shows how meaning is constructed, shared, and learned through social tags upon being contributed by Web users as they perform exploratory searches [47].

Social learning model utilized in collaborative indexing and knowledge exploration for multiple kinds of data is another method [48]. This method is developed based on social learning in collaboration with knowledge representation to offer a collaborative indexing solution for efficient retrieval of semantically related documents. Social tagging system is applied, which allows collaborative indexing of a massive information space based on individual users' subjective interpretation of information. This scheme not only allows better representation of semantics that people can easily interpret, but also lets people with different knowledge backgrounds and information needs share their interpretations of diverse information. Multiple users have personal internal knowledge representations (concepts and mental categories), and they interact with social tagging systems by assigning tags to multiple documents as they use information via systems. The social tagging structure applied in the proposed method facilitates collaborative indexing of a massive information space based on individual users' subjective interpretation of information. Collaborative learning-based indexing technique assists with enhanced representation of semantics and makes them easily interpretable for users who may have different knowledge backgrounds or information needs. Collaborative learning helps with sharing interpretations of different information [48].

Hierarchical tree is an efficient indexing and retrieval approach for human motion data, which is presented on the basis of novel similarity metrics. To provide rapid and accurate retrieval, human motion data is clustered and the relationships between different motion types are identified. Although the technique is suitable for browsing and retrieval of such data, it takes longer for artificial neural network-based unsupervised learning. However, the hierarchical tree yields efficient and accurate results, along with improved performance in comparison with other techniques; clustering accuracy is 98.1 %, and average query retrieval time is 0.24 s [49].

For knowledge management and cooperative work in healthcare networks, Dieng-Kuntz et al. [50] have presented a collaborative-based semantic indexing technique for the reconstitution of a medical ontology. To translate a medical database into an RDF language, natural language processing was applied on a textual dataset. The researchers constructed the 'Virtual Staff' tool, which provides collaborative diagnosis from a number of healthcare members.

Collaborative filtering is implemented to develop knowledge recommendation system which extracts more relevant and precise medical knowledge using keyword search techniques. A recommendation system in medical application is developed based on collaborative filtering to apply realistic search on large volume of medical knowledge and to quickly retrieve accurate knowledge. The system utilizes healthcare team suggestions for medical recommendations and a trust profile is also generated which helps to confidently and abruptly respond to further queries. The mechanism is efficient and appropriate for many medical applications. Thus, the authors claim the effectiveness of collaborative filtering techniques that they are suitable and efficient to develop a knowledge recommendation system. These techniques help in predicting user preferences by analyzing other likely users' preferences as index [51]. Video reindexing is another application of collaborative filtering. The idea of collaborative filtering to develop recommendation system is adopted by Weng and Chuang [52] to improve

the indexing scores, and an unsupervised video reindexing technique is proposed based on this idea. They relate the characteristics of recommendation system with semantic concepts in videos, and user preferences of recommendation system are applied in searching similar videos for user concept. The results show its effectiveness over some supervised learning-based techniques because it uses not only contextual information as other techniques do but it also considers temporal information to retrieve more appropriate video results.

2.3 Datasets used in indexing

Due to the unexpected risks of practical operational systems in real environments, performing real-time indexing is very complex. To provide the big data requirements (i.e., volume, velocity, variety, veracity, value, variability, and complexity) as well as to evaluate indexing accuracy performance, most researchers validate notions by testing in experimental simulated environments that represent real settings [53]. Most NAI techniques use graphs and images for evaluation of indexing accuracy. They also use spatial and textual data in experiments to validate their performance, whereas AI and CAI-based indexing techniques have common behavior to use multimedia data in experimental setup. Along with these type of datasets, AI- based indexing techniques sometimes use textual and annotated data. For example, a Chinese professional baseball league (CPBL) video was utilized as a dataset for assessing indexing techniques [14]. The video includes commentator clips and advertisements that are not considered video, so they are removed from the dataset manually. The normalized dataset covers three video events: home runs, strikeouts, and fly-ball catches. To observe the system's efficiency, an experiment was set up in terms of accuracy rate testing. Table 1 specifies a number of datasets employed by researchers to assess the validity of existing indexing methods. Datasets are identified by their implementation platform or type of data they have. Availability specifies the collection and access process to obtain a dataset for experimentation that whether the dataset is publically available or it is collected specifically for this experiment. In evaluation of some techniques, method of data collection is not specified. Size of each dataset is mentioned in the table to describe volume as the most prominent feature of big data. Evaluation concerns explain the details of preferences in selection of a dataset to test the proposed method. Twenty newsgroups, as an example, contains 20,000 documents of textual data, and these documents are used by Zhu et al. [30] to evaluate the performance of a sparse hashing-based NAI technique. Of these documents, 60 % are taken as training data in development of experiment and remaining 40 % is taken as test data. Image datasets are also added in evaluation process to see the adaptability of technique for other than test data. Furthermore, category in which an indexing technique lies is described in table.

A medical dataset is collected from Nautilius Healthcare and utilized for benchmarking of knowledge management system to work on medical ontology [50]. Medical ontology is developed based on dataset comprising the knowledge and vocabulary which is useful for all actors in healthcare system. Another ontology repository was maintained for the interpretation and storage of a set of cloud services and their natural language descriptions in order to examine the semantic-based indexing. Indexing assists with identifying those cloud resources that satisfy users' needs, and it introduces around 300 different cloud services (e.g., Google Apps, Amazon CloudFront, and Amazon Elastic Beanstalk) [11].

Each of the indexing techniques explained above has some strength that has led researchers to strive to select better techniques for specific domains. For indexing big data, we have identified its requirements in order to appraise these techniques. It is evident that CST is specifically designed for relational database management systems, because it promotes lower implementation cost of Steiner trees when a keyword search is performed [31]. With the identification

Table 1 Datasets used for big data indexing techniques

Indexing category	Data type	Application domain	Availability $(P_1/P_2/U)$ᵃ	Indexing technique	Evaluation concerns	Size	Description
NAI	Image	Image recognition	P_1	Sparse hashing [30]	Approximate similarity search	9298 images in 10 categories	USPS is an image database with 16×16 pixel intensities
					Nearest data neighbor and similar topic retrieval	7291 images as training data	
						2007 images as training data	
		Face recognition			Effectiveness for both large and small datasets	1000 samples as training data	MNIST is a database of face images with 14×14 pixel intensities. The images have different lighting effects, facial expressions, and facial details [54]
						1000 samples as test data	
	Text	Text categorization				7285 documents are split as 72 % training data and 28 % test data	Reuters 21,578 corpus are documents taken from Reuters newswire in 1987. Only 10 categories are considered where documents relate to only one category. It is represented by bag-of-words with a 500-word vocabulary
		Text classification and clustering				18,446 documents are split as 60 % training data and 40 % test data	20 newsgroups data comprise 20,000 newsgroup documents evenly distributed in 20 categories. Dataset is represented as bag-of-words with a 500-word vocabulary

Table 1 continued

Indexing category	Data type	Application domain	Availability ($P_1/P_2/U$)[a]	Indexing technique	Evaluation concerns	Size	Description
	Graph	Biology	P_1	Graph query tree [17]	Efficiency for large and dynamic dataset with high query workload	10–100 K subgraphs are randomly generated	A real dataset is collected from National Cancer Institute which comprises 250 graphs of biological data [55]
					Efficiency of index construction	Average graph size is 10.25	
					Effectiveness of query processing		
	Temporal	Astronomy	P_1	B-tree [33]	Performance of index for different data characteristics	2400 time series, and each time series has 1024 time instances	Mallat Technometrics [56]
			P_1			5000 time series where each time series has 1024 time instances	Light-curve dataset is obtained from Keogh's CD ROM and transformed in piecewise linear line segments with the help of SWAB method [56]
AI	Annotated	ICT	P_1	Semantic [11]	Quality of information retrieval process	300 cloud services are considered for evaluation	This dataset includes semantic annotations of different cloud services (e.g., Google Apps, Amazon Cloud Front, Amazon Elastic Beanstalk, etc.) for semantic-based indexing. These services and their natural language descriptions are inserted in ontology repository
					Performance of single and multi-topic queries	Three services SaaS, PaaS, and IaaS are considered	

Table 1 continued

Indexing category	Data type	Application domain	Availability $(P_1/P_2/U)^a$	Indexing technique	Evaluation concerns	Size	Description
	Temporal	Healthcare	P_2	Self-learning [57]	Correctness of retrieved data for large dataset	Same size dataset with increasing noise (0–50%) Varying size datasets (1000–30,000 in steps of 1000 instances)	This dataset consists of a set of instances, each having five data values: a value for each of the four SIRS parameters and one for whether or not a call was made. A patient makes a call to nurse when he feels system inflammatory response syndrome (SIRS) symptoms and an instance is created
	Multimedia	Sports and games	P_1	State support vector [14]	Relevancy and accuracy of event retrieval for different queries	Video events with 112 home runs, 132 strikeouts, and 138 fly-ball catches 50% of dataset is used as training and 50% as test data	This dataset contains a video standard set of data to be audited, which includes three video events: home runs, strikeouts, and fly-ball catches
CAI	Multimedia (video)	Commerce, research, and Government	P_1	Collaborative filtering [52]	Accuracy of retrieved data	Three large datasets containing 79,484, 18,142, and 35,766 video shots	This dataset is taken from Trecvid official test collections during the years 2006–2008. It includes three sets of video data collected from news and documentaries named as TV06, TV07 and TV07 [58]

Table 1 continued

Indexing category	Data type	Application domain	Availability $(P_1/P_2/U)$[a]	Indexing technique	Evaluation concerns	Size	Description
	Folksonomy	Social tagging	P_2	Collaborative learning [48]	Retrieval of information from relevant documents	Average number of created bookmarks and tags is 90.2 and 425.4 for high-overlap task. Average number of created bookmarks and tags is 42.2 and 212.3 for low-overlap task	An empirical study is conducted where participants with different knowledge backgrounds are asked to search, bookmark, and tag Web pages for two given topics according to their own knowledge representation so that their interpretation of information is obtained. Their concepts are compared with social learning model predictions to see the performance
Cognitive (concept based)	Healthcare		P_2	Collaborative semantic [50]	Quality and relevancy of results	One table has more than 3600 medical terms. One file has a list of root concepts. One table has relations between the terms	A medical database from Nautilius Healthcare is utilized for evaluation purpose which contains knowledge on medicine and conceptual vocabulary of healthcare staff

[a] Availability: public (P_1), private (P_2), and unspecified (U)

of requirements involved in big data indexing, the analysis of available techniques to identify their adaptability becomes easier for researchers.

3 Adopted categorization approach

The current investigation is based on 49 indexing techniques extracted from 60 articles that are related to big data indexing techniques in cloud computing. These articles were selected from credible, highly cited publications and resources, such as ScienceDirect and IEEE based on their impact factor as reported by Journal Citation Reports (JCR). This paper integrates the issues hindering further advancements in big data indexing in cloud computing. It is expected to attract well-respected researchers' consideration to possible solutions with regard to developing indexing techniques for big data in the cloud by analyzing the methodologies' significance in addition to empirical performance.

Analysis is based on key aspects concerning the evaluation and comparison of alternative indexing approaches' performance, that is, the efficiency for volume, velocity, variety, veracity, variability, value, and complexity. The importance of the performance and, especially at this point, the efficiency aspect must be emphasized. As an example, Paul et al. [14] proposed a hybrid of SVM with reinforcement learning agents to identify and organize video events. They claimed that the reinforcement algorithm, if combined with an efficient scheduling scheme, demonstrates significant performance in saving power and time. Their proposed state SVM system indicated 5.30 and 3.34 % improved precision results for three video events of a baseball game dataset. The overall performance gain for precision was 83.83 %, where the query accuracy graph reached 80 %. Weng et al. [52] proposed a collaborative filtering technique in terms of unsupervised video reindexing to improve the detection scores generated by concept classifiers. Latent factor models that use matrix factorizations were applied to remove the inaccurate entries of the pattern score matrix. According to results, the proposed technique improved the detection scores by reducing video reindexing time, especially on multi-core CPUs where matrix factorization for each cluster can be executed independently.

A list of articles is provided as a general overview, with their characteristics and current challenges encumbering non-intelligent and intelligent indexing development in big data in cloud computing. Table 2 comprises three vertical divisions defining indexing classifier types, the authors' work titles, and the objectives. Embedding indexing mechanisms, such as implementing an effective keyword search, optimizes the tree to answer keyword queries, facilitating the discovery of cloud resources to be adapted to cloud computing and make possible the development of efficient indexing mechanisms and reaction systems.

4 Taxonomy of indexing techniques

Taxonomy is an organization which provides a clear understanding of categorization. Based on existing techniques that have been discussed before, we classified them into three categories: NAI, AI, and CAI. Then, taxonomy of indexing techniques is proposed. Figure 1 elaborates the categorization of existing indexing techniques which are analyzed in this survey to see their appropriateness for big data requirements. It presents the detailed taxonomy of indexing techniques and divides these techniques into three main categories such as non-artificial intelligence, artificial intelligence, and collaborative artificial intelligence. The figure also shows the further categorization of each category: NAI is divided as graph-based,

Table 2 List of indexing techniques for big data

Type of indexing technique	Authors	Title of paper	Objectives
B-tree	Li et al. [33]	Top-*k* queries on temporal data	To design a simple and efficient indexing for ranking queries on temporal data
B+-tree	Zhuang et al. [59]	Efficient and robust, large medical image retrieval in the mobile cloud computing environment	To design a medical image retrieval method in mobile cloud computing for searching medical images
Graph partitioning and B+-tree	Sandu Popa et al. [34]	Indexing in-network trajectory flows	To provide efficient indexing for trajectories of moving objects in a network
Composite tree (B-tree)	Wang et al. [10]	High volumes of event stream indexing and efficient multi-keyword searching for cloud monitoring	To design an index for multiple keyword-based queries on generic stream data where bidirectional reference is created between leaf nodes and event indices so that CPU cost is reduced and efficient indexing is achieved
R+-tree	Wei et al. [36]	Indexing spatial data in cloud data management	Presenting a novel multi-dimensional key design index based on an R+-tree KR+ index for efficient search and retrieval of skewed spatial data
R-tree	Wu et al. [60]	A framework for efficient spatial Web object retrieval	To design a hybrid inverted file R-tree to retrieve text and query spatial proximity
Graph query tree	Cheng et al. [17]	Fast graph query processing with a low-cost index	Design a graph querying system that achieves both fast indexing and efficient query processing
Shortest-path tree	Maier et al. [61]	Indexing Network Structure with shortest-path trees	To present and design an indexing technique for auxiliary data structures that provides fast lookups for common operations
Red–Black tree	Yeh et al. [62]	An efficient and secure approach for cloud collaborative editing	Present a red–black tree framework for big data cloud collaborative editing
Steiner tree	Li et al. [31]	Providing built-in keyword search capabilities in RDBMS	To optimize the Steiner tree to answer keyword queries more efficiently, to effectively implement keyword search, and to utilize DBMS capabilities
Authenticated tree-based structures	Li et al. [63]	Authenticated index structures for aggregation queries	To develop efficient index structures for authenticating aggregation queries over large datasets

Table 2 continued

Type of indexing technique	Authors	Title of paper	Objectives
Tree-based structures	Qian et al. [64]	Optimal embedding for shape indexing in medical image databases	To present an optimal shape embedding procedure to index shapes for complete and partial shape similarity retrieval
K-tree	Hsu et al. [65]	Advanced database technologies in a Diabetic Healthcare System	To develop a new indexing method called K-tree to process reverse k-nearest-neighbors (RkNN) queries efficiently
Graph-lattice-based indexing	Yuan and Mitra [66]	Lindex: A lattice-based index for graph databases	To describe indexing techniques based on subgraphs
Bit-sliced index	MacNicol and French [37]	Sybase IQ multiplex-designed for analytics	To present a multi-component bitmap index constructed from three basic encoding schemes
Two-level equality–equality encoding	Sinha and Winslett [67]	Multi-resolution bitmap indexes for scientific data	To propose multi-resolution and parallelizable bitmap indexes
Bitmap	Gündem and Armağan [68]	Efficient storage of healthcare data in XML-based smart cards	To present storage structures for efficiently processing XML path queries on healthcare data
Sparse hashing (SH)	Zhu et al. [30]	Sparse hashing for fast multimedia search	To develop a novel sparse hashing (SH) method for fast approximate similarity searches
Semi-supervised hashing	Wang et al. [69]	Semi-supervised hashing for large-scale search	To develop a technique for nearest neighbor searches using binary codes
Hashing	Ali et al. [70]	Authentication of lossy data in body-sensor networks for cloud-based healthcare monitoring	To design an authentication scheme to detect loss in data using the Merkle hash tree
Hashing	Thilakanathan et al. [71]	A platform for secure monitoring and sharing of generic health data in the Cloud	Design a system to ensure fast healthcare data download using a hash function
Triplet-based hashing	Jayaraman et al. [72]	Use of geometric features of principal components for indexing a biometric database	To propose an indexing technique for a biometric image database
Geometric hashing	Kaushik et al. [73]	An efficient indexing scheme for face database using modified geometric hashing	To present an efficient scheme to index a database of facial images
	Mehrotra et al. [74]	Robust iris indexing scheme using geometric hashing of SIFT key points	To propose an efficient indexing scheme for searching a large iris biometric database

Table 2 continued

Type of indexing technique	Authors	Title of paper	Objectives
HubRank Berkhin's Bookmark Coloring Algorithm (BCA)	Chakrabarti et al. [39]	Index design and query processing for graph conductance search	To design an efficient index for consistent results of PageRank query
Inverted index	Cambazoglu et al. [12]	A term-based inverted index partitioning model for efficient distributed query processing	To minimize the communication overhead that will be incurred by future queries
Permuterm index	Ferragina and Venturini [75]	The compressed permuterm index	To propose a compressed permuterm index which supports fast queries
Three-level indexing hierarchy	Wang et al. [76]	A novel indexing architecture for the provision of smart playback functions in collaborative telemedicine applications	To present a novel indexing architecture in order to support a range of smart playback functions in collaborative telemedicine systems
Lazy indexing	Richter et al. [77]	Towards zero-overhead adaptive indexing in Hadoop	An adaptive indexing approach designed for MapReduce systems at minimal cost
Hierarchical tree	Wu et al. [49]	Indexing and retrieval of human motion data by a hierarchical tree	To develop an efficient indexing and retrieval approach for human motion data
State support vector (SVM) network and reinforcement agent	Paul et al. [14]	Video search and indexing with reinforcement agent for interactive multimedia services	To present a video search and indexing system based on the state support vector (SVM) network, video graph, and reinforcement agent
Fuzzy	Dittrich et al. [46]	MOVIES: Indexing moving objects by shooting index images	To design an indexing technique for such application where objects are moving at a high update rate
Manifold learning	Lazaridis et al. [78]	Multimedia search and retrieval using multi-modal annotation propagation and indexing technique	To propose a complete solution for search and retrieval of rich multimedia content over modern databases
Self-learning	Ongenae et al. [57]	A probabilistic ontology-based platform for self-learning context-aware healthcare applications	To propose a self-learning, probabilistic, ontology-based framework which allows healthcare context-aware applications to adapt their behavior to run-time
Semantic annotation	Rodríguez-García [11]	Creating a semantically enhanced cloud service environment through ontology evolution	To offer a platform to facilitate the retrieval and selection of cloud resources on the basis of keyword search query meeting the users' needs
Randomized interval labeling	Done et al. [79]	Predicting Novel Human Gene Ontology Annotations Using Semantic Analysis	To design a technique to detect Gene Ontology annotations with the help of finding relationships between genes and functions

Table 2 continued

Type of indexing technique	Authors	Title of paper	Objectives
	Yıldırım et al. [80]	GRAIL: A scalable index for reachability queries in very large graphs	To propose randomized interval labeling based on the graph theory
Semantic quad-tree	Zou et al. [81]	Semantic overlay network for large-scale spatial information indexing	To present a novel semantic overlay network for large-scale multi-dimensional spatial information indexing
Phrase-based	Chu et al. [82]	A knowledge-based approach for retrieving scenario-specific medical text documents	To present a new knowledge-based approach to support scenario-specific retrieval applicable in healthcare monitoring
Latent semantic	van der Spek and Klusener [83]	Applying a dynamic threshold to improve cluster detection of LSI	To apply a dynamic threshold to improve cluster detection of latent semantic indexing (LSI)
Semantic audiovisual Web indexing system	Cuggia et al. [84]	Indexing method of digital audiovisual medical resources with semantic Web integration	To propose an audiovisual Web indexing system for medical audiovisual resources
Collaborative social learning	Wai-Tat [48]	Collaborative indexing and knowledge exploration: a social learning model	A machine learning-based approach to present a social learning model which is, in collaboration with knowledge representation, applied as collaborative indexing for retrieval of relevant documents and knowledge exploration
			To present a social learning model which is utilized in collaborative indexing and knowledge exploration
Collaborative unsupervised learning	Weng and Chuang [52]	Collaborative video reindexing via matrix factorization	To present a recommendation system of unsupervised video reindexing developed based on collaborative filtering approach which refines and improves the indexing scores generated by concept classifiers
			To present an unsupervised video reindexing method which refines the detection scores generated by concept classifiers
Collaborative Learning	Huang et al. [51]	Collaboration-based medical knowledge recommendation	To develop a collaborative filtering-based medical knowledge recommendation system so that clinicians can retrieve trust-based accurate knowledge

Table 2 continued

| Type of indexing technique	Authors	Title of paper	Objectives
Collaborative filtering	Komkhao et al. [85]	Incremental collaborative filtering based on Mahalanobis distance and fuzzy membership for recommender systems	To design a model-based collaborating filtering technique to improve the accuracy and scalability of recommender system
Collaborative semantic	Leung and Chan [86]	Semantic music information retrieval using collaborative indexing and filtering	To design a collaborative semantic indexing and metadata-based retrieval for music information so that accurate results are available to users. To design an approach for deep content-based music information retrieval
	Dieng-Kuntz et al. [50]	Building and using a medical ontology for knowledge management and cooperative work in a healthcare network	To present a method for reconstituting a medical ontology by translating a medical database into RDF language in the context of a healthcare network. A virtual staff is developed where more number of healthcare members are involved for better diagnosis
	Elleuch et al. [87]	A fuzzy ontology: based framework for reasoning in visual video content analysis and indexing	To improve the semantic concept detection process through collaboration of fuzzy with ontology
	Gacto et al. [88]	Integration of an index to preserve the semantic interpretability in the multi-objective evolutionary rule selection and tuning of linguistic fuzzy systems	To design an index for natural language context preserving to make it simple and more interpretable
Review	Pandey et al. [89]	An autonomic cloud environment for hosting ECG data analysis services	To design a cloud that collects, stores, and analyzes people's health data
Survey	Graefe [32]	A survey of B-tree locking techniques	To clarify, simplify, and structure the topic of concurrency control in B-trees
Review	Effelsberg [53]	A personal look back at 20 years of research in multimedia content analysis	To present a study of indexing and retrieval of data in the field of multimedia content analysis
Analytical review	Wu et al. [16]	Analyses of multi-level and multi-component compressed bitmap indexes	An analytical comparison of well-known bitmap indexes
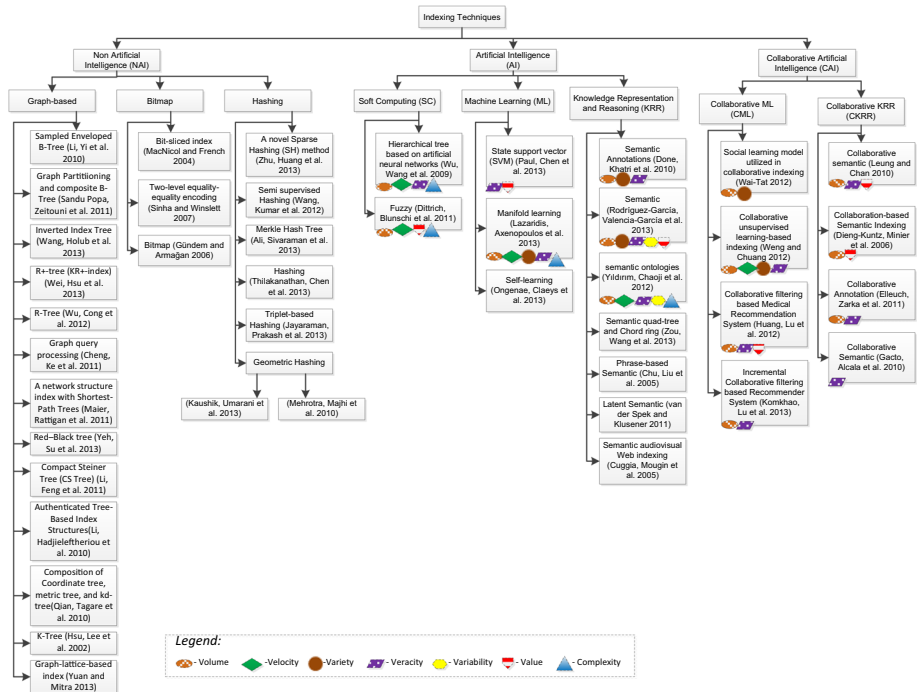
Fig. 1 Taxonomy of indexing techniques

bitmap, and hashing techniques. Similarly, soft computing (SC), machine learning (ML), and knowledge representation and reasoning (KRR) are the sub-categories of AI-based techniques. However for CAI, collaborative ML and collaborative KRR are presented as its sub-categories. Symbols are pasted for such AI and CAI techniques which are examined for our big data indexing requirements. These categories are described as follows:

- *NAI* It involves indexing techniques which are straightforward in index construction and query responses. These methods are mostly developed for rapid and efficient data retrieval. They deploy techniques comprising bitmap, hashing, B-tree, and R-tree and operate as single classifiers of indexing. These techniques are categorized as non-artificial intelligence because rule-based automatic indexing implements only cover-known patterns and they are unable to detect the unknown behavior of big data [90].

- *AI* It relates to highly technical and specialized techniques which utilize knowledge base to provide efficiency in information retrieval. This category amounts to soft computing, machine learning, and knowledge representation and reasoning methods. For instance, fuzzy rule base lies in soft computing AI indexing method which adopts indexing for a large number of moving objects [46]. It is highly efficient in index creation, but as the data is variable, it needs frequent updates in index image. In order to provide up-to-date index, previous index image is discarded which also helps it to be memory efficient. The results show the support of fuzzy rule base to predictive queries and its influence to query response time and update rate. AI-based indexing methods satisfy inconsistent behavior of data and provide solutions to queries related to varying data such as videos [41], human motion [49], multimedia [78], and cloud resources [11]. In cloud resources, a semantic-based artificial

intelligent indexing technique adopts ontology evolution to determine cloud resources for its users.

- *CAI* Indexing techniques based on collaborative artificial intelligence improve accuracy and search efficiency [86]. These techniques consolidate AI techniques to obtain a better cooperative indexing solution. In this category, collaborative machine learning (CML) and collaborative knowledge representation and reasoning (CKRR) methods are included. Collaborative filtering [52], as an example, is an unsupervised CML method which adopts multiple indexing algorithms and collaborates with KRR method to gain high detection rate and for prediction of user missing preferences. The significant advantage of collaborative filtering for indexing is its classifier type independence so it can be applied to any classification results without having to retrain the models.

In NAI techniques, stream data in terms of big data activity is captured by non-artificial intelligence indexing-based classifiers (i.e., graph, B-tree, bitmap, and hashing), whereby online stream data representing desired behavior is analyzed and a behavior model is finally created. Real events take place, current profile is assigned, and an indexing score is computed by comparing the behavior of stream signals. Score normally indicates the degree of irregularity for a specific event, such that the indexing system raises a flag in the event an anomaly occurs when the score surpasses a certain threshold. In this figure, a KR+-index [36] is shown under graph-based indexing techniques of NAI category. The technique is developed using R+-tree to make retrieval process efficient for skewed spatial data on cloud data managements. This is a multi-dimensional indexing technique developed using R+-tree to design a balanced tree so that non-uniformly distributed spatial data becomes evenly distributed on all sites. The index supports scalability of spatial data when compared with other existing techniques. In support of big data requirements, K R+-index performs better for large volume of data.

AI classifiers rely on soft computing (SC), machine learning (ML), and knowledge representation and reasoning (KRR) models, allowing for the patterns analyzed to be categorized. A distinct characteristic of these schemes is the prerequisite for labeled data to train the indexing model, a procedure that places severe demands on resources. ML based on AI classifiers is meant to create an iterative process of observing patterns, adjusting to the mathematical form and making predictions [19]. Manifold learning lies in machine learning which is a sub category of AI- based indexing category of our taxonomy as shown in Fig. 1. A multi-model descriptor index [78] is designed for multimedia data using manifold learning with intent of optimized search and retrieval of large volume data. Symbols in figure show that it supports volume, velocity, variety, veracity, and complexity of big data.

CAI techniques based on multi-agent or non-multi-agent systems enhance the performance of indexing in terms of accuracy and search efficiency. The main objective of CAI consists of collaborative agents in each cluster to provide ML and KRR mechanisms that associate individual and cooperative decision-making to big data indexing in cloud computing. CAI has been employed for big data in cloud computing [91]. Its approaches correspond to a collaborative-based architecture for indexing and modifying a statistical algorithm that measures the similarity between a subject's long-term and short-term behaviors in terms of events. For instance, Weng et al. [52] used collaborative filtering based on video reindexing to detect similarity among mobile users and to predict the missing preferences. It is a concept-based filtering that refines the indexing scores of concept classifiers by exploiting structures embedded within the score matrix. The detection rate is high because various indexing algorithm types are utilized. In addition, the method is independent of classifier type and can be applied to any classification results without having to retrain the models.

Although the collaborative unsupervised method is often more flexible since only the initial scores are required as input, it is not sensitive to big data events and detectable indexing types are limited. Table 3 provides the categorization of techniques according to taxonomy.

In this paper, collaborative-based techniques for big data indexing in cloud computing are contemplated. Figure 2 illustrates the chronology of such indexing techniques that focus on NAI methods (graph-based, hashing, and bitmap), AI methods listed as soft computing (fuzzy set and neural computing), machine learning (support vector machine, manifold learning, and reinforcement learning), knowledge representation and reasoning (semantic ontology), and collaborative ML and collaborative KRR (CAI). The figure also indicates that some techniques derive other techniques to strengthen their design effectiveness and efficiency. Most AI techniques derive graph-based methods to provide better indexing solutions. NAI-based hashing for indexing also utilizes the Merkle tree with network coding to facilitate data verifiability in a lossy environment [70].

Collaborative artificial intelligence methods incorporate ML approaches and, in most cases, KRR approaches. For instance, collaborative indexing methods utilize the semantic algorithm from the KRR category for deep content-based music information retrieval [86]. A collaborative indexing and knowledge exploration mechanism is ML-based method for massive information space based on individual users' subjective interpretations of information from social networks [48]. The main concern is that multiple users have their own internal knowledge representations (concepts and mental categories), and they interact with the social tagging systems by assigning tags to multiple documents as they use information through the system. The two collaborative-based methods were compared against the NAI and AI methods. The summary implicates that indexing accuracy with these methods is extremely high.

5 Comparison of indexing techniques based on big data indexing requirements

The process of indexing is found by applying means such as the NAI method, AI, and CAI-based indexing techniques, which operate as classifiers. This study presents in detail the state-of-the-art in NAI, AI, and CAI systems in the field of indexing big data in cloud computing and additionally highlights the vital concerns/drawbacks to be addressed.

Based on the indexing requirements specified before, the indexing techniques are analyzed. The strengths of each indexing technique are elaborated to ensure its viability for big data. This analysis forms a basis for the development of indexing techniques for such data. Table 4 relates the categorized indexing methods to the presence or absence of volume, velocity, variety, veracity, variability, value, and complexity. These requirements establish the datasets as big datasets.

5.1 Performance of NAI methods

This section outlines the analysis of non-artificial intelligence techniques, comprising graph-based, bitmap, and hashing. Their behavior is emphasized, so their applicability to big data becomes practical.

- B-tree-based indexing scheme supports the largest datasets in terms of number of objects taken from Mallat Technometrics (Mallat in short) and the light-curve datasets. The technique was developed for temporal data, which refers to changing values over time. Along

Table 3 Taxonomy of big data indexing techniques

Method	Performed application domain (C/N/L)[a]	Type of applied dataset (P_1/P_2/U)[b]	Dataset type	Features
Non-artificial intelligent methods (NAI)				
Graph-based				
Tree-based				
Sampled envelope (SE) B-tree for top-k queries [33]	L	P_1/P_2	Temporal	Simple structure of indexing
				Index takes less space
				Its construction is faster
				Small increase in construction cost when variance of data increases
				Faster query response
				Less update cost
Graph partitioning and a composite B+-tree [34]	L	P_1	Trajectory	Faster query response even when query size and data size is increased
				Less update cost which increases gradually
Inverted index tree [10]	C	P_1	Event stream (log)	Index takes less space but takes more time to load in memory
				Manageable query processing cost
				Faster query response
R+-tree (KR+-index) [36]	C	P_1	Spatial	Index takes more space
				Query response time depends upon query size and data size
				Scalable for large data
R-tree [60]	N	P_1	Spatial	Index takes more space
				Query response depends upon buffer size
				Less query processing cost
A graph query processing index system [17]	L	P_1/P_2	Graph	Index takes less space
				Faster index construction
				Faster query response
				Less update cost
				Scalable for large data and query response time remains the same

Table 3 continued

Method	Performed application domain (C/N/L)[a]	Type of applied dataset (P_1/P_2/U)[b]	Dataset type	Features
A network structure index with shortest-path trees [61]	L	P_1/P_2	Graph (network path)	Index takes linear space
				Efficient search for common operations
				Accurate query results
				Applicable on real data sets
				More computational costs for large networks
A framework of Red–Black tree as an efficient and secure approach [62]	C	U	Text	Less index construction cost
				Less update time
				Reduced data encryption overhead
				Efficient encryption compared to 3DES encryption and AES encryption
Compact Steiner tree (CS tree) [31]	L	P_1	Graph	Accurate query results
				Faster query response
Authenticated tree-based index structures [63]	N	P_1/P_2	Spatial	More accurate query results
				Less query execution cost
				Dynamic index updating
Composition of Coordinate tree, metric tree, and *kd*-tree [64]	L	P_1	Image	Less index computational cost
				Fast query response
				Less query execution cost
K-tree [65]	L	P_2	Image	Faster query response
				Accurate query results
A graph-lattice-based index [66]	L	P_1	Graph	Fast query response
				Index construction is faster and easy
				Index update is faster
				Faster query results for subgraph-querying
				False graphs can be filtered easily
Bitmap				
Bit-sliced index [37]	N	U	Transactional	Index takes less space
				Less query processing cost
				Querying is slower than multi-level indexes
Two-level equality-equality encoding [67]	L / N	P_1	Hierarchical data format	Index takes more space
				Faster query response
				Better results for range queries
				Index is scalable in cluster environment

Table 3 continued

Method	Performed application domain (C/N/L)[a]	Type of applied dataset (P_1/P_2/U)[b]	Dataset type	Features
Hashing				
A novel sparse hashing (SH) method [30]	L	P_1/P_2	Image, text	Accurate query results for large datasets
				More index computational cost
				More training cost for large dataset
				Fast encoding
Merkle Hash Tree [70]	C	P_2	Real-time	Accurate query results (90%)
				Less construction cost
Hashing [71]	C	P_2	Medical (ECG)	Efficient query response for large dataset
				More initial setup time
Triplet-based hashing [72]	L	P_1	Medical (ECG)	Index takes less space
				Less computational cost
Geometric hashing				Invariant to scaling
[73]	L	P_1	Image (face)	Index takes less space
				Less computational cost
				Accurate query results
[74]	L	P_1	Image (Iris)	Index takes less space
				Fast query response
				More accurate query results
				Robust in similarity transformations as well as occlusion
				Capable of localizing iris images with change in gaze, occlusion, and illumination
HubRank [39]	L	P_1	Graph	Index takes less space
				Accurate query results
				Less index construction time
				Fast query response
				Efficient query processing
A novel term-based inverted index partitioning model that relies on hypergraph partitioning [12]	N	P_1	Text	Index takes less space
				More computational cost
				Scalable index
A compressed permuterm index (CPI) [75]	L	P_1	Graph	Index takes less space
				Fast query results
				Easy updating

Table 3 continued

Method	Performed application domain (C/N/L)[a]	Type of applied dataset $(P_1/P_2/U)$[b]	Dataset type	Features
Three-level indexing hierarchy (TIH) [76]	L	P_2	Multimedia (video)	Simple index
				Index takes less space
				Less computational cost
				Accurate query results
Artificial intelligent methods (AI)				
Soft computing (SC)				
A hierarchical tree based on artificial neural networks [49]	L	P_1	Motion data	Fast query response
				Accurate query results
				More time-consuming for artificial neural network-based unsupervised learning
Fuzzy [46]	N	P_2	Road Network	Index takes less space
				Less index construction time
				Faster index update
				Faster query response time
				Scalable
				Index image is created frequently so it is time-consuming
Machine learning (ML)				
State support vector (SVM) [14]	L	P_1	Multimedia	Less index construction time
				Accurate query results
				Time-consuming at learning stage
Multimodal descriptor indexing based on manifold learning [78]	L	P_1	Multimedia (Audiovisual)	Less index construction time
				Less index construction cost
				Faster query response
				Scalable
				Time-consuming for manifold learning method
Self-learning [57]	L	P_1	Temporal	Fast query response
				Accurate query results
Knowledge representation and reasoning (KRR)				
Semantic annotations [79]	L	P_1/P_2	Annotated	Accurate query results
Semantic [11]	C	P_1	Annotated	Automatic index updating
				Fast query response
				Accurate query results
				Applicable to unstructured documents

Table 3 continued

Method	Performed application domain (C/N/L)[a]	Type of applied dataset (P_1/P_2/U)[b]	Dataset type	Features
				More information required to provide enough accuracy
				It supports only keyword-based queries
				To ensure accuracy it needs more knowledge
Scalable reachability index (GRAIL) based on semantic ontologies [80]	L	P_1/P_2	Graph	Simple index
				Fast query response for large graphs
				Scalable
				Comparatively low performance for small graphs
Semantic quad-tree and chord ring [81]	N	P_2	Spatial	Scalable
				Supports complex range queries
Phrase-based Semantic [82]	L	P_1	Text	Fast query response in real time
				Accurate query results
Latent semantic [83]	L	P_1	Text	Applicable to large document sets
				Fully automated
Semantic audiovisual Web indexing [84]	N	U	Multimedia (Video)	Simple index
				Demonstrates possibilities of conceptual indexing based on medical ontologies
Collaborative artificial intelligent methods (CAI)				
Collaborative ML				
Social learning model utilized in collaborative indexing [48]	N	P_2	Folksonomy	Faster query response
				Supports structuring of information
				Scalable for large data
				Efficient in semantic representation
				Efficient human–system integration
				Learning is time-consuming
Collaborative unsupervised learning-based indexing via matrix factorization [52]	L	P_1	Multimedia (Video)	Faster index construction
				More query response cost
				Accurate in query results

Table 3 continued

Method	Performed application domain (C/N/L)[a]	Type of applied dataset (P_1/P_2/U)[b]	Dataset type	Features
Collaborative filtering-based medical recommendation system [51]	N or L	P_2	Clinical	Faster query response
				Accurate in query results
				More human effort is required in recommendation recording
				Motivation is required in recording recommendation
Incremental collaborative filtering-based recommender system [85]	L	P_1	Text	More accurate query results
				Scalable and the performance is improved for larger training dataset
Collaborative KRR				
Collaborative semantic [86]	L	U	Multimedia (music)	Accurate in query results
				Accuracy increases as the index is updated
				Index size is gradually increasing
				Fault tolerant
				Resilient, community-validated structure
				Eliminates inappropriate index terms
Collaboration-based semantic indexing [50]	N	P_1/P_2	Cognitive (concept based)	Guaranteed knowledge management
				Useful for a healthcare network dedicated to heavy pathology
				Accurate in query results
Collaborative annotation [87]	L	P_1	Multimedia (video)	Improvement in accuracy of query results
				Improvement in precision of context and concept detection
				More relevant query results
Collaborative semantic [88]	L	P_1	Regression	More accurate query results
				Results are more interpretable

[a] *Performed application domain:* Cloud (C), Network (N), and Local data on a single computer (L)

[b] *Type of applied dataset* Public (P_1), Private (P_2), and Unspecified (U)

with volume and variability, B-tree-based indexing also supports value, as it is feasible for implementation. The required query cost is additionally minimized with this method [33]. To provide indexing for spatial data in cloud data management, R+-tree is scalable and multi-dimensional. The technique based on R+-tree was evaluated for synthetic giga-
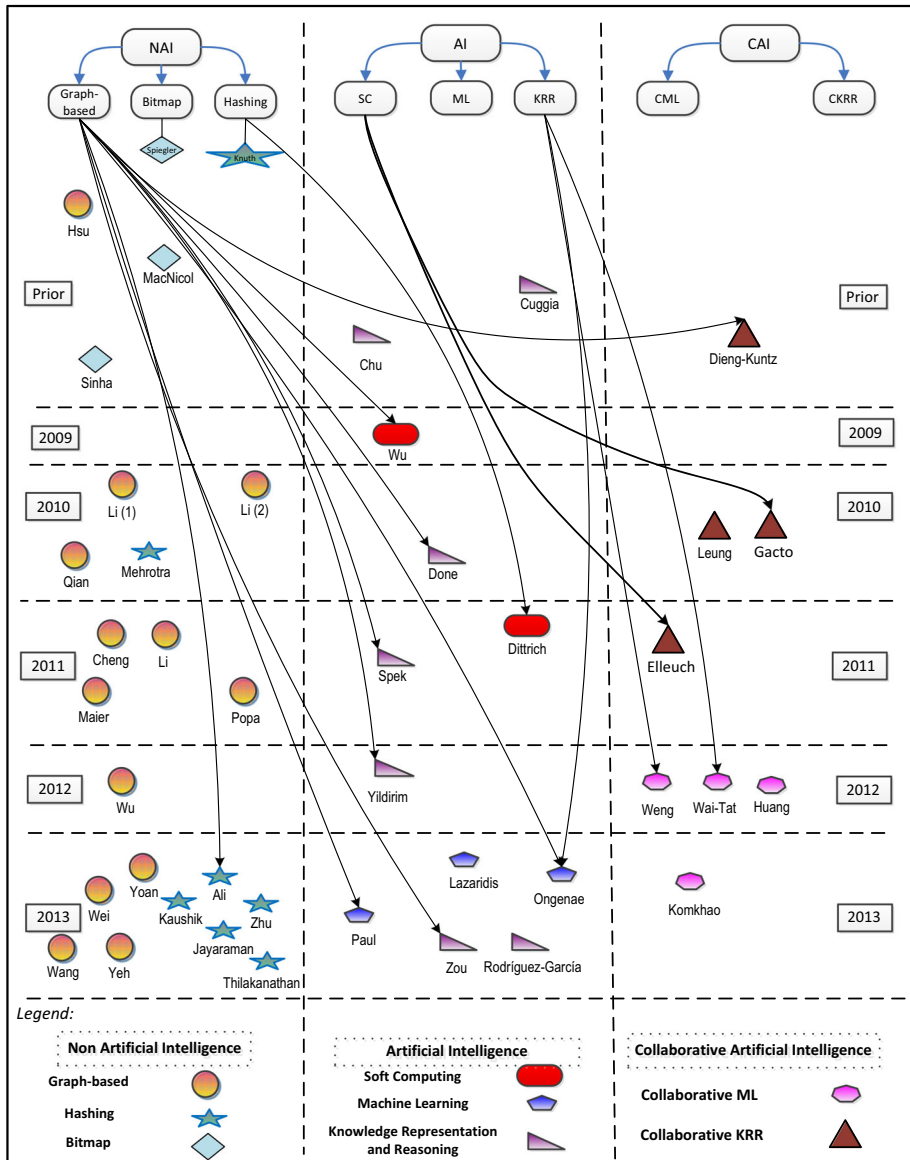
Fig. 2 Chronological order of non-artificial intelligence [NAI], artificial intelligence [AI], and collaborative artificial intelligence [CAI] techniques for big data in cloud computing

byte sets. It indicates better performance for a variable data environment, but since it is specifically designed for spatial data, it does not claim efficiency over other data types [36].

- Bitmap indexing was adopted to satisfy the volume, velocity, variability, and complexity requirements for big data. Two datasets were employed to carry out the evaluation test on this method. From the earth science domain, a data volume of 9.7 GB was collected, having 52 attributes located in 376 HDFS files. Another domain selected for data collection was

Table 4 Analysis of indexing techniques for big data indexing requirements

Indexing method	Authors	Big data indexing requirements[a]						
		Volume	Velocity	Variety	Veracity	Variability	Value	Complexity
Non-artificial intelligence indexing (NAI)								
Graph-based								
B-tree	Li et al. [33]	✓	NA	NA	NA	✓	✓	NA
R+-tree	Wei et al. [36]	✓	NA	×	NA	✓	NA	NA
Suffix tree	Russo et al. [92]	✓	×	NA	NA	NA	NA	NA
Graph Query Tree	Cheng et al. [17]	✓	✓	NA	✓	NA	NA	✓
Shortest Path Tree	Maier et al. [61]	✓	NA	✓	✓	✓	✓	NA
Red–Black tree	Yeh et al. [62]	✓	✓	×	✓	NA	✓	✓
Bitmap	Wu et al. [16]	✓	✓	×	NA	✓	×	✓
Hashing								
Hashing	Zhu et al. [30]	✓	×	✓	×	NA	NA	NA
Geometric hashing	Mehrotra et al. [74]	✓	✓	×	✓	✓	NA	✓
Inverted index	Cambazoglu et al. [12]	✓	NA	×	NA	NA	NA	✓
Lazy indexing	Richter et al. [77]	×	NA	×	✓	NA	✓	NA
Artificial intelligence (AI)								
Semantic indexing	Rodríguez-García et al. [11]	✓	NA	✓	✓	✓	✓	NA
	Done et al. [79]	✓	×	✓	✓	NA	NA	NA
Manifold learning	Lazaridis et al. [78]	✓	✓	✓	✓	NA	NA	✓
Fuzzy	Dittrich et al. [46]	✓	✓	×	×	NA	✓	✓
Support vector machine	Paul et al. [14]	NA	NA	×	✓	NA	✓	NA
Randomized interval labeling	Yıldırım et al. [80]	✓	✓	×	✓	✓	NA	✓
Hierarchical tree	Wu et al. [49]	✓	✓	×	✓	NA	NA	✓

Table 4 continued

Indexing method	Authors	Big data indexing requirements[a]						
		Volume	Velocity	Variety	Veracity	Variability	Value	Complexity
Collaborative artificial intelligence (CAI)								
Collaborative semantic	Leung and Chan [86]	✓	NA	×	✓	NA	✓	NA
	Dieng-Kuntz et al. [50]	✓	NA	×	NA	NA	✓	NA
	Gacto et al. [88]	NA	NA	NA	✓	NA	NA	NA
Collaborative filtering technique	Weng and Chuang [52]	✓	✓	✓	✓	NA	✓	NA
	Huang et al. [51]		NA	×	✓	×	×	×
Incremental collaborative filtering	Komkhao et al. [85]	✓	NA	NA	✓	NA	×	NA
Collaborative learning	Wai-Tat [48]	✓	NA	✓	NA	NA	NA	NA
Collaborative annotation	Elleuch et al. [87]	✓	NA	NA	✓	NA	NA	NA

[a] Big data indexing requirements: ✓ = satisfied, × = not satisfied, NA = not applicable

rocket science, from which 11.1 GB of datasets were gathered. The method proved to be effective for range queries, as it enhances the performance of range queries by a factor of approximately 10. Moreover, index creation speed is also improved. This technique is scalable to cope with changing data volumes and adaptive to data variability. The method illustrates optimal computation performance as $O(h)$, where h is the number of query hits. The datasets chosen for estimation in the proposed bitmap indexing technique do not comprise a variety of data. Evidently, the technique does not support variety of data types [67]. As far as value is concerned, bitmap indexing is expensive to update. It becomes more expensive to update multi-level indexing when multiple indexes are involved [16].

- Hashing technique was applied to design a sparse index for multimedia searches. This method satisfies volume and velocity, whereby better performance was observed for 7285 documents. For evaluation, these documents were split into 72 and 28 % training data and test data, respectively. The actual sizes of these two datasets were 7285 and 18,846. This method is feasible for different sorts of data like optical characters, speech audio, and document scripts. Sparse hashing technique is not suitable for velocity, since it considers fixed data size. Moreover, it does not provide anything related to data accuracy [30].

- A term-based inverted index partitioning method supports volume and complexity for big datasets. Twenty-five million text documents were gathered with 15,417,016 unique terms, and $O(\log K \times \sum_{q_i \in Q} |q_i|^2)$ overall running time complexity was achieved. The method is only feasible for text data; thus, it does not support variety [12]. An indexing scheme for Hadoop systems called Lazy indexing ensures accuracy while adopting piggybacking in one of its phases. Along with ensuring veracity, it also offers high performance benefits for highly selective data block indexing. HDFS datasets were analyzed for performance evaluation, but there was no reference to data volume or variety [77].

5.2 Performance of AI methods

Artificial intelligence methods in further categorization are as SC which incorporates fuzzy and neural computing techniques for data indexing, whereas ML utilizes machine learning and KRR designs semantic-based method. Their feasibility in meeting the challenges of big data is examined as follows.

- Designing an index for video searching based on machine learning attempts to provide accuracy; a precision value of 83.83 % was achieved, and query results for indexing graphs were 80 % accurate. This method does not support data variety since it is designed for video indexing, but the reinforcement agent makes the scheme significant for interactive multimedia indexing [14].

- Randomized interval labeling is an index for very large graphs. The method is analyzed on large, real graphs with an extensive set of experiments, and it is claimed that GRAIL has the ability to scale millions of nodes and edges. It can efficiently handle large graphs and it is a simpler technique for fast and scalable reachability testing. It takes 5791.04 and 27.04 ms for index construction and querying. Since the method was proposed only for large graphs, it does not support other data types. It provides a method to directly maintain exception lists per node, thus eliminating false positives completely. In order to provide evidence for complexity, the authors claim that this mechanism is linear in space and time consumption for indexing. The time for querying ranges from constant to linear according to graph order and size [80].

- Semantic method ensures performance efficiency for large data collection in the cloud. It utilizes datasets from ICT domain knowledge to support this method. For this purpose, up to 300 different services are selected and presented them with this method. Semantic

indexing supports data variety, veracity, variability as well as value. It is practical in dealing with various types of datasets and ontologies. The total average precision value was 88 %, which confirms performance accuracy. This method is also realistic in handling generation of semantic annotations from unstructured documents. Moreover, it is beneficial for up-to-date ontology records [11].

5.3 Performance of CAI methods

Artificial intelligence techniques based on collaborative ML and KRR demonstrate improved indexing accuracy. In this section, we provide the strength of this claim for big data in cloud computing.

- An indexing approach based on collaborative semantic KRR is applied on music information retrieval. To support the evidence of big data requirements, the authors studied the performance for large digital music data. The method ensures a high degree of robustness and fault tolerance as it eliminates inappropriate index terms from the index. This index is dynamically constructed, validated and built up; thus, it demonstrates enhanced performance with maturity of time [86].
- A collaborative filtering technique for videos has three different datasets collected in 2006–2008. Each dataset has a different size and contains different videos. These datasets were analyzed with 374 concepts to support the argument on volume. Performance of 13–52 % was attained when searching concepts inside videos without any other information [52].
- Collaborative learning method presents a social learning model that allows user interpretation-based collaborative indexing for large information sets. The method supports variety, as it is applicable to various types of data [48].

Table 4 presents facts for each indexing category: NAI, AI, and CAI. Suffix tree that supports self-indexing for strings, index construction, and extraction becomes easier using this method. It is evaluated by gathering DNA sequence data spanning 700 MB of space. Efficiency in output reveals the feasibility of this method when it faces large volumes of big data [92]. Other NAI techniques like hashing [30] and inverted index [12] are also capable of handling large datasets.

AI-based indexing, including semantic technique [11] and support vector machine [14], substantiates veracity and value of big data. To validate the semantic technique results, Rodríguez-García et al. [11] obtained up to 300 different services of various data types from the ICT domain. They found 88 % precision value for accuracy. Furthermore, CAI-based indexing [86] ensures volume, veracity, as well as value for robustness of large digital data.

6 Discussion

This paper presents the taxonomy of indexing techniques for big data in cloud computing. Techniques have mostly common strengths and weaknesses within a category, and they support some of our defined big data requirements. NAI-based indexing techniques are mostly used owing to their efficiency in index construction and index sizes. They take very less time in index construction, and the index has smaller size. It is common for these methods to satisfy large amounts of data, which is a basic requirement of big data. However, they do not support various data types and data formats.

In Tables 5 and 6, contemporary AI and CAI-based indexing techniques are presented and analyzed, respectively, on the basis of their performance efficiency. Moreover, their corre-

lation with big data challenges is also kept in view to better justify them. These techniques provide superior indexing solutions in terms of query results and execution accuracy, which form the basis of our analysis. These techniques are evaluated with the previously defined indexing requirements for big data. Furthermore, the strengths and weaknesses are also taken into account for their evaluation. Three levels (i.e., high, medium, and low) were established to report the performance of each indexing method provided for this top selection based on the analysis in previous sections.

Artificial intelligence-based indexing methods display high detection accuracy in most cases. Other than query result accuracy, they also demonstrate time efficiency in query response and index creation. Responses for querying are faster, and index creation is not time taking. These methods improve not only search performance but also search result accuracy. They are scalable and adaptable for large indexing volume requirements. Like NAI techniques, these techniques do not support variety of data. It is observed that most AI techniques are not time efficient in the learning stage. Their link with big data challenges highlights the fact that AI-based techniques concentrate more on veracity and volume of data but overlook design properties to index a variety of data. Other challenges are satisfied on average.

State support vector machine and reinforcement agent for video event indexing performs better only for range queries, but for other data types, it is not so efficient [14]. This technique is feasible to obtain accurate results from queries and the efficiency in index construction is also notable, but the support to big data requirements is average. Manifold learning [78] is another machine learning-based indexing solution for audiovisual data which exhibit almost similar features to the prior, but it has more support toward big data requirements and that is why graded with high level in efficiency. KRR-based semantic method [11] for indexing cloud resources is a method that supports heterogeneous data, and its variability along with other challenges such as volume, veracity, and value is settled by AI techniques. For this reason, it is advisable to choose a semantic method for indexing big data. Fuzzy-based SC indexing [46] provides indexing for videos for which an index needs to be updated more frequently. This updating process decreases efficiency. Otherwise, the technique is efficient in index creation and generating rapid query responses. Furthermore, fuzzy-based indexing supports large volumes and velocity of video data, which are major concerns in indexing. Manifold learning [78] is also a high-level ML-based indexing scheme developed for indexing large multimedia datasets. It provides better support to a variety of data.

As far as CAI methods are concerned, they take into account detection rate as well as accuracy of query results. The majority of CAI-based indexing techniques is accomplished to satisfy large volume data challenges and provides more accuracy of query results. Their evidence in favor of velocity and value of data is very low, whereas for other requirements, it is average. In comparison with the performance of AI-based indexing techniques for the fulfillment of big data requirements, these techniques endeavor quiet better results in terms of volume and veracity. In discussion of big data, the challenge of handling large amount of data is the main consideration, whereas for measuring performance of indexing techniques, accuracy and efficiency in query responses are imperative judgment points. Thus, it can be undoubtedly claimed that CAI techniques are the better choice as foothold to develop an indexing technique specifically for big data.

Collaborative filtering [52] implements an unsupervised method to refine indexing results, and overall, it can be argued that it is superior due to its detection performance. Big data challenges such as large volume, velocity, variety, and veracity are satisfied for indexing video data. Rather, it presents overall enhanced indexing performance yet higher computational cost, which categorizes it as medium. Another collaborative filtering-based approach [51]

Table 5 Performance of AI-based indexing techniques

Type of indexing technique	Authors	Method	Strengths	Weaknesses	Requirements		Level
					Satisfied	Unsatisfied	
SVM	Paul et al. [14]	State support vector machine and reinforcement agent is implemented for identification and management of video events	Less index construction time Accurate query results	Time-consuming at learning stage	Veracity Value	Variety	Low
Fuzzy	Dittrich et al. [46]	Fuzzy-based index captures images from moving objects to create an index	Index takes less space Less index construction time Faster index update Faster query response time Scalable	Index image is created frequently so it is time-consuming	Volume Velocity Value Complexity	Variety Veracity	Medium
Manifold Learning	Lazaridis et al. [78]	For optimized search and retrieval of multimedia content manifold learning adopts a multi-modal search for large scaling indexing	Less index construction time Less index construction cost Faster query response Scalable	Time-consuming for manifold learning method	Volume Velocity Variety Veracity Complexity		High
Randomized interval labeling	Yıldırım et al. [80]	A KRR method based on the graph theory to provide indexing for large graphs	Simple index Fast query response for large graphs	Comparatively low performance for small graphs	Volume Velocity	Variety	High

Table 5 continued

Type of indexing technique	Authors	Method	Strengths	Weaknesses	Requirements Satisfied	Unsatisfied	Level
			Scalable		Veracity Variability Complexity		
Hierarchical Tree	Wu et al. [49]	Clustering of human motion data and identification of motion types are used to provide efficient indexing and retrieval	Fast query response Accurate query results	More time- consuming for artificial neural network-based unsupervised learning	Volume Velocity Veracity Complexity	Variety	Medium
Semantic	Rodríguez-García et al. [11]	Through ontology evolution, the semantic index module provides support to identify cloud resources according to user needs	Automatic index updating Fast query response Accurate query results Applicable to unstructured documents	It supports only keyword-based queries To ensure accuracy it needs more knowledge	Volume Variety Veracity Variability Value		High
	Done et al. [79]	A vector space model for semantic analysis is applied in information retrieval to extend latent semantic indexing for Gene Ontology	Accurate query results		Volume Variety Veracity	Velocity	Low

Table 6 Performance of CAI-based indexing techniques

Type of indexing technique	Authors	Method	Strengths	Weaknesses	Requirements		Level
					Satisfied	Unsatisfied	
Collaborative filtering technique	Weng and Chuang [52]	An unsupervised method to reindex or refine the results of concept classifier is implemented for videos. Collaborative filtering is applied to refine the results	High detection rate It is applicable for each classifier type	More computational resources are required Expensive technique	Volume Velocity Variety Veracity		Medium
	Huang et al. [51]	A collaborative filtering-based knowledge recommendation system which uses clinicians experience to retrieve accurate and reliable medical knowledge	More accurate and relevant results Improved reliability if more recommendations are considered	Human effort is required to insert recommendations	Volume Veracity Value	Variety Variability	Medium
Incremental collaborative filtering	Komkhao et al. [85]	Model-based collaborative filtering approach is applied so that models are built continuously by grouping users into clusters so that search becomes simple and accurate	More accurate query results Scalable and the performance is improved for larger training dataset		Volume Veracity	Value	Medium

Table 6 continued

Type of indexing technique	Authors	Method	Strengths	Weaknesses	Requirements		Level
					Satisfied	Unsatisfied	
Collaborative learning	Wai-Tat [48]	Collaborative indexing and knowledge exploration are used to represent the semantic	It allows people to share their interpretations according to their background knowledge	Learning time-consuming	Volume Variety		Low
Collaborative semantic	Leung and Chan [86]	Semantic algorithm is used in collaborative indexing for music information retrieval. Subtle nuances and emotional impressions are used for information retrieval	Robust and fault-tolerant approach which discards inappropriate index terms	Support to only keyword-based queries	Volume Veracity Value	Variety	High
	Dieng-Kuntz et al. [50]	The CAI-based technique is used to reestablish medical ontology where the concept of 'Virtual Staff' is introduced to identify and diagnose with the help of healthcare members	Efficient for large medical ontology Allows real-time updates and the history of therapeutic decisions	Limited to keyword search queries	Volume Value	Variety	Low
	Gacto et al. [88]	The index applies multi-objective evolutionary algorithm for rule selection and tuning of membership functions to improve accuracy of results while preserving the semantic interpretability of database	More accurate query results Results are more interpretable		Veracity		Low

Table 6 continued

Type of indexing technique	Authors	Method	Strengths	Weaknesses	Requirements		Level
					Satisfied	Unsatisfied	
Collaborative annotation	Elleuch et al. [87]	A fuzzy ontology-based method to improve the detection of semantic knowledge of video indexing system. Concepts of multimedia are extracted for this purpose	Improvement in accuracy of query results Improvement in precision of context and concept detection More relevant query results		Volume Veracity		Medium

which suggests a medical recommendation system to retrieve accurate knowledge is also a better adaptable CAI approach. Evaluation on real dataset collected from different clinicians proved its trust worthiness on retrieved knowledge, but for the sake of accuracy, more clinicians' recommendations are needed to be inserted in system. The approach satisfies large volume, veracity, and value but does not support variety and variability of data. Furthermore, yet no evidence is found respective to velocity, and the approach is rated as medium level. Incremental collaborative filtering approach [85] is a model-based method which is scalable and shows improved indexing performance when larger dataset is available. Like other CAI-based techniques, it is also proving accuracy of querying results and large size data disregarding other requirements.

Collaborative semantic method [86] is robust and fault tolerant for music information retrieval. It supports large datasets and provides accuracy of query results with high value, deeming it a high-level indexing method. Annotation method which lies in KRR category when collaborates with fuzzy ontology [87] achieves high accuracy of query results for large volume of multimedia data. Progress in precision and context detection for more relevant results is depicted by it. So the collaborative annotation method satisfies volume and veracity requirements for multimedia data.

7 Conclusion

A categorization of existing data management and indexing techniques for big data has been outlined in this paper. The main objective was to analyze the indexing requirements for big data as well as to present state-of-the-art potential indexing techniques to provide researchers a basis for developing enhanced solutions in a specific domain in order to provide a support to heterogeneity, accuracy, and scalability of data as major concern. Presentation of issues inhibiting progress in big data is also a considerable aspect of this study. This work predominantly focuses on collaborative artificial intelligence techniques accuracy of querying in information retrieval. In taxonomy, indexing techniques were assessed and classified as NAI, AI, and CAI. Published research papers on the implementation of these indexing techniques were studied and analyzed for big data indexing requirements. Furthermore, important aspects of each technique were derived by critically reviewing these papers, which is meant to lead researchers to progress in many applications of big data indexing. A discussion was organized accordingly to present the survey and show the capability of each technique regarding big data indexing requirements. To conclude the discussion, it can be said that CAI-based indexing methods are more potent for data indexing and retrieval, since they are adaptable to large size data, which is the main issue for BD-MCC. CAI-based methods provide satisfactory retrieval rate and accuracy of data retrieval in cloud, whereby data are continuously captured and utilized by end users. Owing to their adaptability, CAI-based indexing methods can be implemented in future to satisfy other big data indexing requirements such as veracity, variability, value, and complexity.

References

1. Gärtner M, Rauber A, Berger H (2013) Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation. Knowl Inf Syst 1–32. doi:10.1007/s10115-013-0678-y
2. Demirkan H, Delen D (2013) Leveraging the capabilities of service-oriented decision support systems: putting analytics and big data in cloud. Decis Support Syst 55(1):412–421. doi:10.1016/j.dss.2012.05.048
3. Amer-Yahia S, Doan A, Kleinberg J, Koudas N, Franklin M (2010) Crowds, clouds, and algorithms: exploring the human side of "big data" applications. Paper presented at the proceedings of the 2010 ACM SIGMOD international conference on management of data, Indianapolis, Indiana, USA
4. Dixon Z, Moxley J (2013) Everything is illuminated: what big data can tell us about teacher commentary. Assess Writ 18(4):241–256. doi:10.1016/j.asw.2013.08.002
5. Liu W, Peng S, Du W, Wang W, Zeng GS (2014) Security-aware intermediate data placement strategy in scientific cloud workflows. Knowl Inf Syst 41:1–25
6. Dopazo J (2013) Genomics and transcriptomics in drug discovery. Drug Discov Today 19(2):126–132. doi:10.1016/j.drudis.2013.06.003
7. Wang J, Wu S, Gao H, Li J, Ooi BC (2010) Indexing multi-dimensional data in a cloud system. In: Proceedings of the 2010 ACM SIGMOD international conference on management of data. ACM, pp 591–602
8. Fiore S, D'Anca A, Palazzo C, Foster I, Williams DN, Aloisio G (2013) Ophidia: toward big data analytics for escience. Proc Comput Sci 18:2376–2385. doi:10.1016/j.procs.2013.05.409
9. Chen J, Chen Y, Du X, Li C, Lu J, Zhao S, Zhou X (2013) Big data challenge: a data management perspective. Front Comput Sci 7(2):157–164. doi:10.1007/s11704-013-3903-7
10. Wang M, Holub V, Murphy J, O'Sullivan P (2013) High volumes of event stream indexing and efficient multi-keyword searching for cloud monitoring. Future Gener Comput Syst 29(8):1943–1962
11. Rodríguez-García MÁ, Valencia-García R, García-Sánchez F, Samper-Zapater JJ (2013) Creating a semantically-enhanced cloud services environment through ontology evolution. Future Gener Comput Syst 32:295–306. doi:10.1016/j.future.2013.08.003
12. Cambazoglu BB, Kayaaslan E, Jonassen S, Aykanat C (2013) A term-based inverted index partitioning model for efficient distributed query processing. ACM Trans Web 7(3):1–23. doi:10.1145/2516633.2516637
13. Bast H, Celikik M (2013) Efficient fuzzy search in large text collections. ACM Trans Inf Syst 31(2):1–59. doi:10.1145/2457465.2457470
14. Paul A, Chen B-W, Bharanitharan K, Wang J-F (2013) Video search and indexing with reinforcement agent for interactive multimedia services. ACM Trans Embed Comput Syst 12(2):1–16. doi:10.1145/2423636.2423643
15. Kadiyala S, Shiri N (2008) A compact multi-resolution index for variable length queries in time series databases. Knowl Inf Syst 15(2):131–147
16. Wu K, Shoshani A, Stockinger K (2010) Analyses of multi-level and multi-component compressed bitmap indexes. ACM Trans Database Syst 35(1):1–52. doi:10.1145/1670243.1670245
17. Cheng J, Ke Y, Fu AW-C, Yu JX (2011) Fast graph query processing with a low-cost index. VLDB J 20(4):521–539
18. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47. doi:10.1145/505282.505283
19. Shamshirband S, Anuar NB, Kiah MLM, Patel A (2013) An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique. Eng Appl Artif Intell 26(9):2105–2127. doi:10.1016/j.engappai.2013.04.010
20. Fan C-Y, Chang P-C, Lin J-J, Hsieh JC (2011) A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification. Appl Soft Comput 11(1):632–644. doi:10.1016/j.asoc.2009.12.023
21. Chang RM, Kauffman RJ, Kwon Y (2014) Understanding the paradigm shift to computational social science in the presence of big data. Decis Support Syst 63:67–80. doi:10.1016/j.dss.2013.08.008
22. Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Ullah Khan S (2015) The rise of "big data" on cloud computing: review and open research issues. Inform Syst 47:98–115. doi:10.1016/j.is.2014.07.006
23. Katal A, Wazid M, Goudar RH (2013) Big data: issues, challenges, tools and good practices. In: 2013 Sixth international conference on contemporary computing (IC3), 2013, pp 404–409. doi:10.1109/IC3.2013.6612229

24. Kaisler S, Armour F, Espinosa JA, Money W (2013) Big data: issues and challenges moving forward. In: 2013 46th Hawaii international conference on system sciences (HICSS), 2013, pp 995–1004. doi:10.1109/HICSS.2013.645

25. Yang C, Zhang X, Zhong C, Liu C, Pei J, Ramamohanarao K, Chen J (2014) A spatiotemporal compression based approach for efficient big data processing on Cloud. J Comput Syst Sci 80(8):1563–1583. doi:10.1016/j.jcss.2014.04.022

26. Philip Chen C, Zhang C-Y (2014) Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf Sci 275:314–347

27. Wang X, Luo X, Liu H (2014) Measuring the veracity of web event via uncertainty. J Syst Softw 1–11. doi:10.1016/j.jss.2014.07.023

28. LaValle S, Lesser E, Shockley R, Hopkins MS, Kruschwitz N (2013) Big data, analytics and the path from insights to value. MIT Sloan Manag Rev 21:21–31

29. Barbierato E, Gribaudo M, Iacono M (2014) Performance evaluation of NoSQL big-data applications using multi-formalism models. Future Gener Comput Syst 37:345–353. doi:10.1016/j.future.2013.12.036

30. Zhu X, Huang Z, Cheng H, Cui J, Shen HT (2013) Sparse hashing for fast multimedia search. ACM Trans Inf Syst 31(2):1–24. doi:10.1145/2457465.2457469

31. Li G, Feng J, Zhou X, Wang J (2011) Providing built-in keyword search capabilities in RDBMS. VLDB J 20(1):1–19

32. Graefe G (2010) A survey of B-tree locking techniques. ACM Trans Database Syst 35(3):16

33. Li F, Yi K, Le W (2010) Top-k queries on temporal data. VLDB J 19(5):715–733

34. Sandu Popa I, Zeitouni K, Oria V, Barth D, Vial S (2011) Indexing in-network trajectory flows. VLDB J 20(5):643–669

35. Sellis TK, Roussopoulos N, Faloutsos C (1987) The R+-tree: a dynamic index for multi-dimensional objects. Paper presented at the proceedings of the 13th international conference on very large data bases

36. Wei L-Y, Hsu Y-T, Peng W-C, Lee W-C (2013) Indexing spatial data in cloud data managements. Pervasive Mobile Comput 1–14. doi:10.1016/j.pmcj.2013.07.001

37. MacNicol R, French B (2004) Sybase IQ multiplex-designed for analytics. Paper presented at the proceedings of the thirteenth international conference on very large data bases, vol 30, Toronto, Canada

38. Shang L, Yang L, Wang F, Chan K-P, Hua X-S (2010) Real-time large scale near-duplicate web video retrieval. In: Proceedings of the international conference on multimedia, 2010. ACM, pp 531–540

39. Chakrabarti S, Pathak A, Gupta M (2011) Index design and query processing for graph conductance search. VLDB J 20(3):445–470. doi:10.1007/s00778-010-0204-8

40. Wang Y (2008) On contemporary denotational mathematics for computational intelligence. In: Gavrilova ML, Kenneth Tan CJ, Wang Y, Yao Y, Wang G (eds) Transactions on computational science II. Springer, Berlin, pp 6–29

41. Chen-Yu C, Ta-Cheng W, Jhing-Fa W, Li Pang S (2009) SVM-based state transition framework for dynamical human behavior identification. In: IEEE international conference on acoustics, speech and signal processing, 2009. ICASSP 2009, pp 1933–1936. doi:10.1109/ICASSP.2009.4959988

42. Ohbuchi R, Kobayashi J (2006) Unsupervised learning from a corpus for shape-based 3D model retrieval. Paper presented at the proceedings of the 8th ACM international workshop on multimedia information retrieval, Santa Barbara, CA, USA

43. Saul LK, Roweis ST (2003) Think globally, fit locally: unsupervised learning of low dimensional manifolds. J Mach Learn Res 4:119–155. doi:10.1162/153244304322972667

44. He J, Li M, Zhang H-J, Tong H, Zhang C (2004) Manifold-ranking based image retrieval. Paper presented at the proceedings of the 12th annual ACM international conference on multimedia, New York, NY, USA

45. Bordogna G, Pagani M, Pasi G (2006) A dynamic hierarchical fuzzy clustering algorithm for information filtering. In: Herrera-Viedma E, Pasi G, Crestani F (eds) Soft computing in web information retrieval. Springer, Berlin, pp 3–23

46. Dittrich J, Blunschi L, Vaz Salles M (2011) MOVIES: indexing moving objects by shooting index images. Geoinformatica 15(4):727–767. doi:10.1007/s10707-011-0122-y

47. Dillenbourg P, Järvelä S, Fischer F (2009) The evolution of research on computer-supported collaborative learning. In: Balacheff N, Ludvigsen S, de Jong T, Lazonder A, Barnes S (eds) Technology-enhanced learning. Springer, Berlin, pp 3–19

48. Wai-Tat F (2012) Collaborative indexing and knowledge exploration: a social learning model. IEEE Intell Syst 27:39–46

49. Wu S, Wang Z, Xia S (2009) Indexing and retrieval of human motion data by a hierarchical tree. Paper presented at the proceedings of the 16th ACM symposium on virtual reality software and technology, Kyoto, Japan

50. Dieng-Kuntz R, Minier D, Růžička M, Corby F, Corby O, Alamarguy L (2006) Building and using a medical ontology for knowledge management and cooperative work in a health care network. Comput Biol Med 36(7–8):871–892. doi:10.1016/j.compbiomed.2005.04.015

51. Huang Z, Lu X, Duan H, Zhao C (2012) Collaboration-based medical knowledge recommendation. Artif Intell Med 55(1):13–24

52. Weng M-F, Chuang Y-Y (2012) Collaborative video reindexing via matrix factorization. ACM Trans Multimed Comput Commun Appl 8(2):23

53. Effelsberg W (2013) A personal look back at twenty years of research in multimedia content analysis. ACM Trans Multimed Comput Commun Appl 9(1s):43

54. The ORL Database of Faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. Accessed 31 Oct 2014

55. Data set of NCI. http://discover.nci.nih.gov/datasets.jsp. Accessed 31 Oct 2014

56. Keogh E, Xi X, Wei L, Ratanamahatana C (2006) The UCR time series dataset. http://www.cs.ucr.edu/~eamonn/time_series_data/

57. Ongenae F, Claeys M, Dupont T, Kerckhove W, Verhoeve P, Dhaene T, De Turck F (2013) A probabilistic ontology-based platform for self-learning context-aware healthcare applications. Expert Syst Appl 40(18):7629–7646. doi:10.1016/j.eswa.2013.07.038

58. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and TRECVid. Paper presented at the proceedings of the 8th ACM international workshop on multimedia information retrieval, Santa Barbara, CA, USA

59. Zhuang Y, Jiang N, Wu Z, Li Q, Chiu DK, Hu H (2013) Efficient and robust large medical image retrieval in mobile cloud computing environment. Inf Sci 263:60–86. doi:10.1016/j.ins.2013.10.013

60. Wu D, Cong G, Jensen CS (2012) A framework for efficient spatial web object retrieval. VLDB J 21(6):797–822

61. Maier M, Rattigan M, Jensen D (2011) Indexing network structure with shortest-path trees. ACM Trans Knowl Discov Data 5(3):15

62. Yeh S-C, Su M-Y, Chen H-H, Lin C-Y (2013) An efficient and secure approach for a cloud collaborative editing. J Netw Comput Appl 36(6):1632–1641. doi:10.1016/j.jnca.2013.05.012

63. Li F, Hadjieleftheriou M, Kollios G, Reyzin L (2010) Authenticated index structures for aggregation queries. ACM Trans Inf Syst Secur 13(4):1–35. doi:10.1145/1880022.1880026

64. Qian X, Tagare HD, Fulbright RK, Long R, Antani S (2010) Optimal embedding for shape indexing in medical image databases. Med Image Anal 14(3):243–254. doi:10.1016/j.media.2010.01.001

65. Hsu W, Lee ML, Ooi BC, Mohanty PK, Teo KL, Xia C (2002) Advanced database technologies in a diabetic healthcare system. Paper presented at the proceedings of the 28th international conference on very large data bases, Hong Kong, China

66. Yuan D, Mitra P (2013) Lindex: a lattice-based index for graph databases. VLDB J 22(2):229–252. doi:10.1007/s00778-012-0284-8

67. Sinha RR, Winslett M (2007) Multi-resolution bitmap indexes for scientific data. ACM Trans Database Syst 32(3):16. doi:10.1145/1272743.1272746

68. Gündem Tİ, Armağan Ö (2006) Efficient storage of healthcare data in XML-based smart cards. Comput Methods Programs Biomed 81(1):26–40. doi:10.1016/j.cmpb.2005.10.007

69. Wang J, Kumar S, Chang S (2012) Semi-supervised hashing for large scale search. IEEE Trans Pattern Anal Mach Intell 34(12). doi:10.1109/TPAMI.2012.48

70. Ali ST, Sivaraman V, Ostry D (2013) Authentication of lossy data in body-sensor networks for cloud-based healthcare monitoring. Future Gener Comput Syst 35:80–90. doi:10.1016/j.future.2013.09.007

71. Thilakanathan D, Chen S, Nepal S, Calvo R, Alem L (2013) A platform for secure monitoring and sharing of generic health data in the Cloud. Future Gener Comput Syst 35:102–113. doi:10.1016/j.future.2013.09.011

72. Jayaraman U, Prakash S, Gupta P (2013) Use of geometric features of principal components for indexing a biometric database. Math Comput Model 58(1–2):147–164. doi:10.1016/j.mcm.2012.06.005

73. Kaushik VD, Umarani J, Gupta AK, Gupta AK, Gupta P (2013) An efficient indexing scheme for face database using modified geometric hashing. Neurocomputing 116:208–221. doi:10.1016/j.neucom.2011.12.056

74. Mehrotra H, Majhi B, Gupta P (2010) Robust iris indexing scheme using geometric hashing of SIFT keypoints. J Netw Comput Appl 33(3):300–313. doi:10.1016/j.jnca.2009.12.005

75. Ferragina P, Venturini R (2010) The compressed permuterm index. ACM Trans Algorithms 7(1):1–21. doi:10.1145/1868237.1868248

76. Wang C-H, Jiau HC, Chung P-C, Ssu K-F, Yang T-L, Tsai F-J (2010) A novel indexing architecture for the provision of smart playback functions in collaborative telemedicine applications. Comput Biol Med 40(2):138–148

77. Richter S, Quiané-Ruiz J-A, Schuh S, Dittrich J (2012) Towards zero-overhead adaptive indexing in Hadoop. arXiv preprint arXiv:12123480

78. Lazaridis M, Axenopoulos A, Rafailidis D, Daras P (2013) Multimedia search and retrieval using multimodal annotation propagation and indexing techniques. Sig Process Image Commun 28(4):351–367. doi:10.1016/j.image.2012.04.001

79. Done B, Khatri P, Done A, Draghici S (2010) Predicting novel human gene ontology annotations using semantic analysis. IEEE/ACM Trans Comput Biol Bioinform 7(1):91–99

80. Yıldırım H, Chaoji V, Zaki M (2012) GRAIL: a scalable index for reachability queries in very large graphs. VLDB J 21(4):509–534. doi:10.1007/s00778-011-0256-4

81. Zou Z, Wang Y, Cao K, Qu T, Wang Z (2013) Semantic overlay network for large-scale spatial information indexing. Comput Geosci 57:208–217. doi:10.1016/j.cageo.2013.04.019

82. Chu WW, Liu Z, Mao W, Zou Q (2005) A knowledge-based approach for retrieving scenario-specific medical text documents. Control Eng Pract 13(9):1105–1121. doi:10.1016/j.conengprac.2004.12.011

83. van der Spek P, Klusener S (2011) Applying a dynamic threshold to improve cluster detection of LSI. Sci Comput Program 76(12):1261–1274. doi:10.1016/j.scico.2010.12.004

84. Cuggia M, Mougin F, Beux PL (2005) Indexing method of digital audiovisual medical resources with semantic Web integration. Int J Med Inform 74(2–4):169–177. doi:10.1016/j.ijmedinf.2004.04.027

85. Komkhao M, Lu J, Li Z, Halang WA (2013) Incremental collaborative filtering based on Mahalanobis distance and fuzzy membership for recommender systems. Int J Gen Syst 42(1):41–66

86. Leung CHC, Chan WS (2010) Semantic music information retrieval using collaborative indexing and filtering. In: Gelenbe E, Lent R, Sakellari G, Sacan A, Toroslu H, Yazici A (eds) Computer and information sciences, vol 62. Lecture notes in electrical engineering. Springer, Netherlands, pp 345–350. doi:10.1007/978-90-481-9794-1_65

87. Elleuch N, Zarka M, Ammar AB, Alimi AM (2011) A fuzzy ontology: based framework for reasoning in visual video content analysis and indexing. Paper presented at the proceedings of the eleventh international workshop on multimedia data mining, San Diego, CA, USA

88. Gacto MJ, Alcala R, Herrera F (2010) Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems. IEEE Trans Fuzzy Syst 18(3):515–531. doi:10.1109/TFUZZ.2010.2041008

89. Pandey S, Voorsluys W, Niu S, Khandoker A, Buyya R (2012) An autonomic cloud environment for hosting ECG data analysis services. Future Gener Comput Syst 28(1):147–154

90. van Zuylen H (2012) Artificial intelligence applications to critical transportation issues. Transportation Research E-Circular, Transportation Research Board, pp 3–5

91. Doelitzscher F, Reich C, Knahl M, Passfall A, Clarke N (2012) An agent based business aware incident detection system for cloud environments. J Cloud Comput 1(1):1–19

92. Russo LM, Navarro G, Oliveira AL (2008) Fully-compressed suffix trees. In: LATIN 2008: Theoretical informatics. Springer, Berlin, pp 362–373

Abdullah Gani is Professor at the Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His academic qualifications were obtained from the University of Hull, UK for bachelor and master degrees, and the University of Sheffield, UK for Ph.D. He has vast teaching experience due to having worked in various educational institutions locally and abroad—schools, teaching college, ministry of education, and universities. His interest in research started in 1983 when he was chosen to attend the Scientific Research Course in REC-SAM by the Ministry of Education, Malaysia. More than 100 academic papers have been published in conferences and respectable journals. He actively supervises many students at all level of study—Bachelor, Master and Ph.D. His interest of research includes self-organized system, reinforcement learning, wireless-related networks. He is now working on mobile cloud computing with High Impact Research Grant of USD 500,000 (RM 1.5 M) for the period of 2011–2016. He is a senior member of IEEE. Currently, he is a director of the Centre for Mobile Cloud Computing Research, which focuses on high impact research. He is also a visiting Professor at the King Saud University, Saudi Arabia as well as serves as Adjunct Professor at the COMSATS Institute of Information Technology, Islamabad, Pakistan.

Aisha Siddiqa is a Ph.D. candidate and researcher in Faculty of Computer Science and Information Technology, University of Malaya. She graduated as Bachelor of Science (Information Technology) from Bahauddin Zakariya University Multan, Pakistan and obtained her Masters of Science (Computer Science) degree from COMSATS Institute of Information Technology Islamabad, Pakistan. Her area of research is Algorithms and Data Structures, Big Data Storage, Big Data Analysis and mainly Indexing.

Shahaboddin Shamshirband is a research fellow at the Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His academic qualifications were obtained from the Islamic Azad University, Sari and Mashhad for bachelor and master degrees, and the University of Malaya (2014), Malaysia for Ph.D. His interest of research includes game theory, reinforcement learning, and wireless-related networks. He is an editorial board and act as a reviewer for various top journals with ISI indexed. He is a member of IEEE.

Fariza Hanum graduated with Bachelors of Science (Computer Science) and Masters of Science (MIS) from Northern Illinois University, USA, and later obtained her Ph.D. from University of Malaya. She has worked in the industry as Systems Analyst for ten years before joining the academia in 1997. She is currently serving as a Senior Lecturer at the Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya. She became involved in multi-disciplined research but her main focuses are in databases, information systems, and in data sciences.