Data Sci. Eng. (2016) 1:265–284
DOI 10.1007/s41019-016-0022-0

Big Data Reduction Methods: A Survey

Muhammad Habib ur Rehman[1] · Chee Sun Liew[1] · Assad Abbas[2] · Prem Prakash Jayaraman[3] · Teh Ying Wah[1] · Samee U. Khan[2]

Received: 7 August 2016 / Revised: 18 November 2016 / Accepted: 18 November 2016 / Published online: 10 December 2016

Abstract Research on big data analytics is entering in the new phase called fast data where multiple gigabytes of data arrive in the big data systems every second. Modern big data systems collect inherently complex data streams due to the volume, velocity, value, variety, variability, and veracity in the acquired data and consequently give rise to the 6Vs of big data. The reduced and relevant data streams are perceived to be more useful than collecting raw, redundant, inconsistent, and noisy data. Another perspective for big data reduction is that the million variables big datasets cause the curse of dimensionality which requires unbounded computational resources to uncover actionable knowledge patterns. This article presents a review of methods that are used for big data reduction. It also presents a detailed taxonomic discussion of big data reduction methods including the network theory, big data compression, dimension reduction, redundancy elimination, data mining, and machine learning methods. In addition, the open research issues pertinent to the big data reduction are also highlighted.

Keywords Big data · Data compression · Data reduction · Data complexity · Dimensionality reduction

✉ Muhammad Habib ur Rehman
mhrehman@siswa.um.edu.my

[1] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

[2] Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND, USA

[3] Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, Australia

1 Introduction

Big data is the aggregation of large-scale, voluminous, and multi-format data streams originated from heterogeneous and autonomous data sources [1]. The volume is the primary characteristic of big data that is represented by the acquisition of storage spaces in large-scale data centers and storage area networks. The massive size of the big data not only causes the data heterogeneity but also results in diverse dimensionalities in the datasets. Therefore, efforts are required to reduce the volume to effectively analyze big data [2]. In addition, big data streams are needed to be processed online to avoid lateral resource consumption for storage and processing. The second key characteristic of big data is velocity. The velocity refers to the frequency of data streams, which is needed to be abridged in order to handle big data effectively. For example, solar dynamics observatory generates excess of one terabytes data per day and the analysis of such a fast big data is possible only after reduction or summarization [3]. On the other hand, big data inherits the 'curse of dimensionality.' In other words, millions of dimensions (variables, features, attributes) are required to be effectively reduced to uncover the maximum knowledge patterns [4, 5]. For example, behavior profiles of the Internet users that mainly comprise of searches, page-views, and click-stream data are sparse and high dimensional with millions of possible keywords and URLs [6]. Similarly, personal genomic high-throughput sequencing not only increases the volume and velocity of data but also adds to the high dimensionality of the data [7]. Therefore, it is imperative to reduce the high dimensions while retaining the most important and useful data.

Data reduction methods for big data vary from pure dimension reduction techniques to compression-based data reduction methods and algorithms for preprocessing, cluster-level data deduplication, redundancy elimination, and implementation of network (graph) theory concepts. Dimension reduction techniques are useful to handle the heterogeneity and massiveness of big data by reducing million variable data into manageable size [8–11]. These techniques usually work at post-data collection phases. Similarly, cluster deduplication and redundancy elimination algorithms that remove duplicated data for efficient data processing and useful knowledge discovery are primarily post-data collection methods [12–15]. Recently, the network theory concepts have also been employed for big data reduction [16–18]. The aforementioned methods first extract the semantics and linked structures from the unstructured datasets and then apply graph theory for network optimization. Conversely, some methods to reduce big data during the data collection process are also proposed in the recent literature [19–21]. In this study, we presented a detailed discussion of these data reduction methods.

This article presents a thorough literature review of methods for big data reduction. A few similar prior studies have also been conducted. However, these studies either present a generic discussion of big data reduction or discuss a specific group of relevant systems or methods. For example, the authors in [1] discussed the big data reduction to be the critical part of mining sparse, uncertain, and incomplete data. Similarly, the authors in [22, 23] argue big data reduction as the critical part of data analysis and data preprocessing. However, both of the studies lack in presenting discussion about specific systems and methods for big data reduction. The authors in [4] discussed big data reduction issue specifically by focusing on dimension reduction, whereas the authors in [24] emphasized on the data compression. However, a wide range of methods remain unexplored. Currently, there is no specific study in the literature that addresses the core issue of big data reduction. Therefore, we aim to present a detailed literature review that is specifically articulated to highlight the existing methods relevant to big data reduction. In addition, some open research issues are also presented to direct future researchers.

The main contributions of this article are:

- A thorough literature review and classification of big data reduction methods are presented.
- Recently proposed schemes for big data reduction are analyzed and synthesized.
- A detailed gap analysis for the articulation of limitations and future research challenges for data reduction in big data environments is presented.

The article is structured as follows: Sect. 2 discuses the complexity problem in big data and highlights the importance of big data reduction. The taxonomical discussion on big data reduction methods is presented in Sect. 3. The discussion on open issues and future research challenges is given in Sect. 4, and finally, the article is concluded in Sect. 5.

2 Big Data Complexity and the Need for Data Reduction

Big data systems include social media data aggregators, industrial sensor networks, scientific experimental systems, connected health, and several other application areas. The data collection from large-scale local and remote sensing devices and networks, Internet-enabled data streams, and/or devices, systems, and networks-logs brings massively heterogeneous, multi-source, multi-format, aggregated, and continuous big data streams. Effectively handling the big data stream to store, index, and query the data sources for lateral data processing is among the key challenges being addressed by researchers [25, 26]. However, data scientists are facing data deluge issue to uncover the maximum knowledge patterns at fine-grained level for effective and personalized utilization of big data systems [3, 27]. The data deluge is due to 6Vs properties of big data, namely the volume, variety, value, velocity, veracity, and variability. The authors in [26] discussed the 6Vs as follows.

- *Volume* The data size characterizes the volume of big data. However, there is no agreed upon definition of big data which specifies the amount of data to be considered as 'big' on order to meet the definition of big data. However, a common sense is developed in research community who consider any data size as big in terms of volume which is not easily processable by underlying computing systems. For example, a large distributed system such as computing clusters- or cloud-based data centers may offer to process multiple terabytes of data but a standalone computer or resource constrained mobile devices may not offer the computational power to process even a few gigabytes of data. Therefore, the volume property of big data varies according to underlying computing systems.
- *Velocity* The velocity of big data is determined by the frequency of data streams which are entering in big data systems. The velocity is handled by big data systems in two ways. First, the whole data streams are collected in centralized systems, and then, further data processing is performed. In the second approach, the data streams are processed immediately after data collection before storing in big data systems. The second approach is more practical; however, it requires a lot of programming efforts and computational resources in order to reduce and filter the data streams before entering in big data systems.

- *Variety* Big data systems collect data stream from multiple data sources which produce data streams in multiple formats. This heterogeneity in data sources and data types impacts the variety property-related characteristics. Therefore, big data systems must be able to process multiple types of data stream in order to effectively uncover hidden knowledge patterns.
- *Veracity* The utility of big data systems increases when the data streams are collected from reliable and trustworthy sources. In addition, the data stream collection is performed with compromising the quality of data streams. The veracity property of big data relates to reliability and trustworthiness of big data systems.
- *Variability* Since all data sources in big data systems do not generate the data streams with same speed and same quality. Therefore, variability property enables to handle the relevant issues. For example, the elastic resource provisioning as per the requirements of big data systems.
- *Value* The value property of big data defines the utility, usability, and usefulness of big data systems. This property tends more toward the outcomes of data analytics and data processing processes and is directly proportional to other 5Vs in big data systems.

The well-designed big data systems must able to deal with all 6Vs effectively by creating a balance between data processing objectives and the cost of data processing (i.e., computational, financial, programming efforts) in big data systems.

Moreover, the complexity in big data systems emerges in three forms: (1) data complexity, (2) computational complexity, and (3) system complexity [28]. The data complexity arises due to multiple formats and unstructured nature of big data, which elevate the issue of multiple dimensions and the complex inter-dimensional and intra-dimensional relationships. For example, the semantic relationship between different values of the same attribute, for example, noise level in the particular areas of the city, increases the inter-dimensional complexity. Likewise, the linked relationship among different attributes (for example, age, gender, and health records) raises the intra-dimensional complexity issue. In addition, the increasing level of data complexity in any big data system is directly proportional to the increase in computational complexity where only the sophisticated algorithms and methods can address the issue. Moreover, the system-level complexity is increased due to extensive computational requirements of big data systems to handle extremely large volume, complex (mostly unstructured and semi-structured), and sparse nature of the data. The extensive literature review exhibits that the big data reduction methods and systems have potential to deal with the big data complexity at both algorithms and systems level. In addition to data complexity, the big data reduction problem is studied in various other perspectives to articulate the effects and the need of data reduction for big data analysis, management, commercialization, and personalization.

Big data analysis also known as big data mining is a tedious task involving extraneous efforts to reduce data in a manageable size to uncover maximum knowledge patterns. To make it beneficial for data analysis, a number of preprocessing techniques for summarization, sketching, anomaly detection, dimension reduction, noise removal, and outliers detection are applied to reduce, refine, and clean big data [29]. The New York Times, a leading US newspaper, reports that data scientists spend 50–80% of the time on cleaning the big datasets [30]. The terms used in the industry for the aforementioned process are ‘data munging,’ ‘data wrangling,’ or ‘data janitor work.’ Another issue with the large-scale high-dimensional data analysis is the over-fitting of learning models that are generated from large numbers of attributes with a few examples. These learning models fit well within the training data, but their performance with testing data significantly degrades [31].

Data management is another important aspect to discuss the big data reduction problem. The effective big data management plays a pivotal role from data acquisition to analysis and visualization. Although data acquisition from multiple sources and aggregation of relevant datasets improve the efficiency of big data systems, it increases the in-network processing and data movement at clusters and data center levels. Similarly, the indexing techniques discussed in [26] enhance the big data management; however, the techniques come across data processing overheads. Although the conversion of unstructured data to semi-structured and structured formats is useful for effective query execution, the conversion in itself is a time- and resource-consuming activity. Moreover, big data is huge in volume that is distributed in different storage facilities. Therefore, the development of learning models and uncovering global knowledge from massively distributed big data is a tedious task. Efficient storage management of reduced and relevant data enhances both the local learning and global view of the whole big data [32, 33]. Currently, visual data mining technique of selecting subspace from the entire feature spaces and subsequently finding the relevant data patterns also require effective data management techniques. Therefore, the reduction in big data at the earliest enhances the data management and data quality and therefore improves the indexing, storage, analysis, and visualization operations of big data systems.

Recently, businesses particularly the enterprises are turning into big data systems. The collection of large data streams from Web users’ personal data streams (clickstreams, ambulation activities, geo-locations, and health records) and integration of those data streams with personalized services is a key challenge [34]. The collection of irrelevant data streams increases the computational burden that directly affects the operational cost of enterprises. Therefore, the collection of fine-grained, highly relevant, and reduced data streams from users is another challenge that requires serious attention while designing big data systems. Currently, user data collection by third parties without explicit consent and information about commercialization is raising the privacy issues. The participatory personal data where users collect and mine their own data and participate for further utilization and customization of services in ubiquitous environments can address the issue of fine-grained data availability for enterprises. Keeping in view the big data complexity, the need for big data reduction, and analyzing big data reduction problem in different perspective, we present a thorough literature review of the methods for big data reduction.

The core technological support for big data reduction methods is based on multilayer architecture (see Fig. 1). The data storage is enabled by large-scale data centers and networks of different computing clusters [35]. The storage infrastructures are managed by core networking services, embarrassingly parallel distributed computing frameworks, such as Hadoop map-reduce implementations and large-scale virtualization technologies [36]. In addition, cloud services for the provision of computing, networking, and storage are also enabled using different cloud-based operating systems. A recent phenomenon in cloud computing is enabling the edge-cloud services by the virtualization of core cloud services near the data sources. Recently, Cisco released a Fog cloud to enable the intercommunication between core cloud services and proximal networks of data sources [37, 38]. At the lowest layers of the big data architecture resides the multi-format data sources which include standalone mobile devices, Internet-enabled social media data streams, remotely deployed wireless sensor networks, and large-scale scientific data streams among many others. This layered architecture enables to process and manage big data at multiple levels using various computing systems with different form factors. Therefore, wide ranges of application models are designed and new systems have been developed for big data processing.

3 Big Data Reduction Methods

This section presents the data reduction methods being applied in big data systems. The methods either optimize the storage or in-network movement of data or reduce data redundancy and duplication. In addition, some of the methods only reduce the volume by compressing the original data and some of the methods reduce the velocity of data streams at the earliest before entering in big data storage systems. Alternatively, some of the methods extract topological structures of unstructured data and reduce the overall big data using network theory approaches that are discussed as follows.

3.1 Network Theory

Network (also known as graph) theory is playing a primary role in reduction of high-dimensional unstructured big data into low-dimensional structured data [39]. However, the extraction of topological structures (networks) from big data is quite challenging due to the heterogeneity and complex data structures. The authors in [40] proposed network theory-based approach to extract the topological and dynamical network properties from big data. The topological networks are constructed by establishing and evaluating relationships (links) among different data points. The statistical node analysis of the networks is performed for optimization and big data reduction [41]. The optimized networks are represented as small-world networks, free-scale networks, and random networks and are ranked on the basis of statistical parameters, namely mean, standard deviation, and variance. Mathematically, scale-free networks are formally represented as given in Eq. 1 using the main parameter as shown in Eq. 2.

$$p_k \approx k^{-\gamma} \tag{1}$$

$$\gamma_{p_k} = -\frac{\log p_k}{\log k} \tag{2}$$

where p_k represents fraction of nodes having k degree and parameter γ having range of $2 < \gamma < 3$.

Similarly, formal representation of random networks is presented in Eq. 3 using the main parameter as shown in Eq. 4.

$$p^k \approx \frac{z^k e^{-z}}{k!} \tag{3}$$

$$\log \frac{p_k k!}{(n-1)^k} = k \cdot \log(p) + (1-n)p \tag{4}$$

where p is the probability distribution of edges between any two nodes, n shows the number of nodes and z is calculated as $z = (n-1)p$. The mathematical representation of small-world networks is performed using Eq. 5 with main parameter as shown in Eq. 6.

$$d \propto \log(n) \tag{5}$$

$$d = \propto \log(n) \tag{6}$$

where n represents the nodes in network and d is the distance between two randomly chosen nodes in the network.

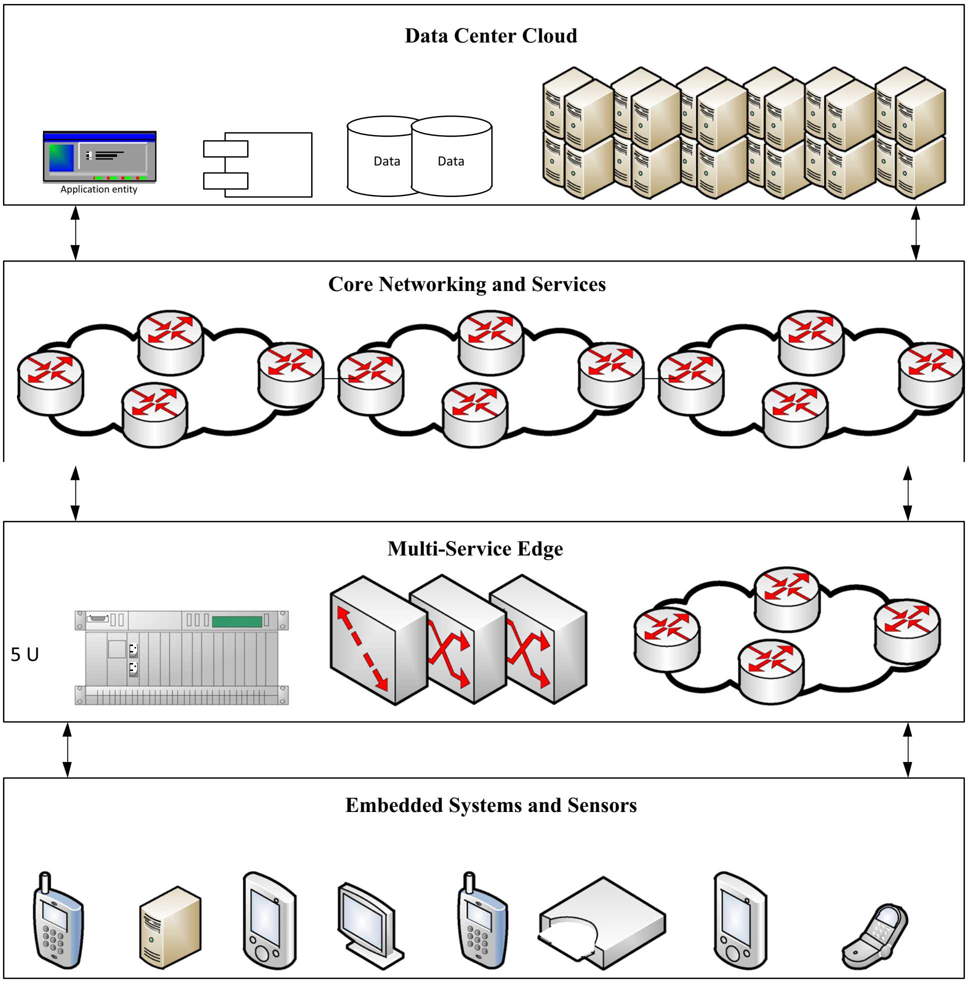


Fig. 1 Multilayer architecture for big data systems

Traditionally, causality analysis is performed to establish the connection between different nodes. However, the extraction of topological networks from unstructured data is hard due to high dependency and one-way relationship on the links. The alternate to causality analysis is the assessment of influence among different nodes and establishing the connection between them on the basis of their co-occurrence. Although the technique does not ensure that nodes in the network can influence each other, the assessment of their co-occurrence could become a strong relationship parameter to link different nodes. Preferential attachment property enables influence assessment [41]. The property exhibits that newly established connections are most probably made with highly connected nodes in the topological networks. In addition, the identification of influential nodes and effective assessment of their co-occurrence can significantly reduce big data. Mathematically, the co-occurrence-based influence measure $\Delta(v_{i1}, v_{il})$ between two nodes v_{i1} and v_{il} is represented as shown in Eq. 7.

$$\Delta(v_{i1}, v_{il}) = \frac{1}{(|p(v_{i1}, v_{il})| - 1)(x-a)^n \sum_{p_i \in p(v_{i1}, v_{il})} W_{p_i}} \times \left[\hat{L}_{(v_{i1}, v_{il})} \sum_{i \neq j = 1}^{|p(v_{i1}, v_{il})|} \times \left(\prod_{i=2}^{l-2} W(v_i, v_{i+1}) + \prod_{j=2}^{l-2} W(v_j, v_{j+1}) \right) \right] \tag{7}$$

where $W_{p_i} = \frac{w(v_{i1}, v_{i2})}{deg(v_{i1})}$ and $p(v_{i1}, v_{il})$ is the shortest path between v_{i1} and v_{il} and $0 \leq \Delta(v_{i1}, v_{il}) \leq 1$.

Aside from the influence-based assessment, similarity graph and collapsing simplicial complexes in network structures are used for topological network reduction in big datasets. Similarity graph is used to model the relationship among different datasets on the basis of their similarity matrix [17]. Further optimization of similarity graph is performed by merging every vertex with the maximal similarity clique (MSC), where a similarity clique (SC) is the collection of nodes with negligible distance. The notion is to convert a SC into MSC by adding an adjacent vertex that violates the properties of the SC. The graph is reduced when every MSC in the similarity graph is merged into a single node and the number of nodes and their edges are reduced in turn. Although the MSC is an efficient scheme to reduce topological networks, finding all of the MSCs in the similarity graph is quite challenging and computationally inefficient.

Moreover, the simplicial complexes are the multi-dimensional algebraic representations of large datasets [18]. The simplicial complex approach is applied over persistent homology, which is a method to compute topological features of large spaces at different levels of spatial resolutions. The persistent homology algorithm, as an alternate of hierarchical clustering, takes nested sequences of simplicial complexes and returns the summary of topological features at all levels. The concept of strong collapse that is keeping relevance information about all of the nodes in the simplicial complex is used to reduce the big data. The strong collapse is also useful for parallel and distributed computations, but the increasing computational complexities of maintaining relevance information of all of the nodes in complex prove to be the bottleneck. The concept of selective collapse algorithm is introduced to reduce the computational complexity of processing persistent homology. The proposed framework keeps computation traces of collapsing the complex using persistence algorithm proposed in [42]. In addition, the strong collapses are represented by forest that facilitates in easy processing of persistence across all nodes of the complex. Although the selective collapse algorithm is useful for the big data reduction, the empirical evidences are still lacking in the literature.

3.2 Compression

The reduced-size datasets are easy to handle in terms of processing and in-network data movement inside the big data storage systems. Compression-based methods are suitable candidates for data reduction in terms of size by preserving the whole data streams. Although computationally inefficient and involving decompression overhead, the methods allow to preserve the entire datasets in the original form. Numerous techniques for big data compression are proposed in the literature, including spatiotemporal compression, gzip, anamorphic stretch transform (AST), compressed sensing, parallel compression, sketching, and adaptive compression, to name a few [43–45]. Table 1 presents the summary of these methods.

Big data reduction in cloud environments is quite challenging due to multiple levels of virtualization and heterogeneity in the underlying cloud infrastructure. A spatiotemporal technique for data compression on big graph data in the cloud generates reduced datasets [45]. The technique performs online clustering of streaming data by correlating similarities in their time series to share workload in the clusters. In addition, it performs temporal compression on each network node to reduce the overall data. The proposed technique effectively meets the data processing quality and acceptable fidelity loss of the most of the application requirements. On the other hand, wireless sensor networks (WSNs) are generating large data streams at massive scales. The spatiotemporal data compression algorithm proposed in [46] ensures efficient communication, transmission, and storage of data in WSNs-based big data environment. The proposed approach not only reduces the size of transmitted data but also ensures prolonged network lifetime. The algorithm measures the correlation degree of sensed data, which determines the content of the data to be transmitted.

The authors in [44] proposed an efficient big data reduction scheme for the IP-activity dataset in social science. The techniques are based on compression method to utilize the standalone computer machines instead of large-scale distributed systems, such as Hadoop and big table. The authors used eight core processors from a 32 core AMD Opteron Processor 5356 machine of 2.3 GHZ speed and 20 GB RAM. The 9 TB of loose text files was converted into a binary format for readability in social science settings. The methodology is based on three steps for: (1) information representation, (2) parallel processing, and (3) compression and storage. Big data reduction is performed in second and third steps. First, each of the data files is processed and converted into a corresponding HDFS file; then, map-reduce approach is used to link each individual HDFS file with the corresponding geographical locations and aggregate in lateral stages. In map-phase, the files are linked, whereas in the reduce stage, all of the resultant HDFS files are converted into a single HDFS file. Finally, the gzip algorithm was used with the compression level of five over the optimal data chunk size of 100,000 × 1. The results exhibited a significant amount of data reduction from 9.08 TB (raw data) to 3.07 TB (HDFS converted data) and 0.50 TB (compressed data). Although the proposed approach contributed significantly, it is only useful for the given dataset. Therefore, online data analysis for big data reduction remains challenging in streaming big data environments.

The anamorphic stretch transform (AST) stretches the sharp features more strongly as compared to the coarse features of the [47]. The AST could be applied to both the analogue (for digitization) and digital signals (for compression). The technique is primarily based on self-adaptive stretch where more samples are associated with sharp features and fewer samples are associated with redundant coarse features. The strength of the AST is its ability to enhance the utility of limited samples as well as reducing the overall size of the data. The results demonstrated that

Table 1 Big data compression methods

References	Methods	Description	Strengths	Weaknesses
Yang et al. [45]	Spatiotemporal	The proposed method performs online clustering of streaming data by correlating similarities in their time series to share workload in the clusters. In addition, it performs temporal compression on each network node to reduce overall data	Performance enhancement in terms of data quality, information fidelity loss, and big data reduction	At least one time processing of whole data is performed in cloud environment which increases the operational cost
Ackermann and Angus [44]	gZip	gZip is a compression tool developed that is being used for big data reduction to improve the resource efficiency for the IP-activity dataset in social science	It provides a light weight and simple file format; therefore, it has low computational complexity	gZip compresses one file at a time; therefore, massively parallel programming models like map-reduce must be used for performance gain in terms of computation time
Jalali and Asghari [47]	AST	The AST is a novel method that is used to compress digital signal by performing selective stretching and wrapping methods. The technique is primarily based on self-adaptive stretch where more samples are associated with sharp features and less samples are associated with redundant coarse features	AST performs data compression of the signal extracted on frequency domain. The method also performs inverse transformation of the constructed signal	The method specifically works with big data involving signal processing. The generalization to other domain is a bottleneck in this research
Wang et al. [48]	Compressed sensing	Compressed sensing is a compressible and/or sparse signal that projects a high-dimensional data in low-dimensional space using random measurement matrix	The proposed scheme performs data acquisition and compression in parallel for improved performance as compared with Nyquist sampling theory-based compression methods	The probability of poor data quality and information fidelity loss increases when the analyses are performed on reduced and compressed data
Brinkmann et al. [50]	RED encoding	RED encoding used to manage massively generated voluminous electrophysiology data	RED performs best when encoding invariant signals, and it provides high compression rate with improved computational speed in lossless compression of time series signals	The performance of the RED encoding methods degrades with high variance in signals
Bi et al. [55]	Parallel compression	Parallel compression methods uses proper orthogonal decomposition method to compress data in order to effectively trace and extract useful features from the data	Balances between feature retention error and compression ratio Performs fast decompression for interactive data visualization	Due to noise, the standard deviation of error remains high in the dataset
Monreale et al. [46]	Sketching	Sketching uses count-min sketch algorithm to compress vehicular movement data and achieve compact communication	Guarantees data reduction and preserves some important characteristics of the original data	The probability of information fidelity loss is more when sketching applied with inconsistent and noisy data stream

56-fold data could be compressed using the AST equations (see Eq. 8 and Eq. 9).

$$I(\omega) = \mathrm{AST}\{\tilde{E}(\omega)\} = \int_{-\infty}^{+\infty} \tilde{E}(\omega)\tilde{E}*(\omega+\omega_{\mathrm{m}})e^{j[\varphi(\omega)-\varphi(\omega+\omega_{\mathrm{m}})]}\mathrm{d}\omega \qquad (8)$$

where $(\omega+\omega_{\mathrm{m}})$ and ω_{m} represent carrier and modulation frequencies, respectively. The $\varphi(\omega)$ is an auto-correlation function of the AST with embedded kernel that represents a frequency-dependent phase operation. The compression or expansion of time, bandwidth product is dependent on the correct selection of $\varphi(\omega)$. The structured modulation distribution function, S_{M}, is used to select $\varphi(\omega)$.

$$S_{\mathrm{M}}(\omega_{\mathrm{m}}, t) = \int_{-\infty}^{+\infty} \tilde{E}(\omega)\tilde{E}*(\omega+\omega_{\mathrm{m}})e^{j[\varphi(\omega)-\varphi(\omega+\omega_{\mathrm{m}})]}e^{j\omega t}\mathrm{d}\omega \qquad (9)$$

where t represents the time variable.

The storage of big data and its complete processing for anomaly detection raises the issues of privacy due to exposure of each data point. The authors proposed compressed sensing also known as compressed sampling technique for compressible and/or sparse signals that project a high-dimensional data in low-dimensional space using random measurement matrix [48]. The method is selected because compressive sensing theory enables to address the limitations of Nyquist sampling [49] theory and can perform data acquisition and compression simultaneously. This strength of the compressed sensing theory enables to detect anomalies in big data streams. Mathematically, for a signal, $x \in R^N$ compressed sensing can be represented using Eq. 10.

$$x = \sum_{i-1}^{N} \emptyset_i \theta_i \text{ or } x = \emptyset\theta \tag{10}$$

where $\emptyset$ shows the $N \times N$ orthonormal transform basis and θ is used as the expansion coefficient vector under the orthonormal basis. If $x \in R^K$ sparse signal where $K \neq 0$ in vector θ and $K < N$, the signal x can be collected with a small set of non-adaptive and linear measurements (see Eq. 11).

$$y = \Psi x = \Psi\emptyset\theta \tag{11}$$

where Ψ is a $M \times N$ random measurement matrix and $M < N$. Here, $(\emptyset, \Psi)$ represents a pair of orthobases that follow the incoherence restriction.

Compressive sensing theory is useful because low-dimensional space fully represents high-dimensional data. The results showed that the proposed algorithm produced satisfactory results with and without compressed sensing. The compression was applied by the ratio of 1:3 and 1:5, and the experiments for human detection were performed from behind the brick and gypsum walls.

The authors in [50] proposed a novel file format called multi-scale electrophysiology format (MEF) to manage massively generated electrophysiology data. The block-wise lossy compression algorithm (RED encoding) is used in addition to the cyclic redundancy check (CRC), encryption, and block index structure of the MEF. The RED encoding ensures high lossless compression rate and higher computational speed and is also able to adopt with statistical variation in the raw data, which is very important for non-stationary ECG signals. The experimental results showed that for 32 KHZ recordings with each block of 32,556 samples acquired in one second, the MEF obtained reasonably better compression rate. The authors recommend recording at least 2000 samples in each block for the maximum performance gain.

The amount of data generated in vehicular ad hoc networks is massive due to on-board monitoring of vehicles and relevant spatiotemporal data acquisition. The authors in [46] utilized sketching algorithm called count-min sketch algorithm to compress vehicular movement data and achieved compact communication. The algorithm maps frequency counters to compressed vectors using hash tables. Although the main focus of the proposed study is on privacy preservation, data reduction is also performed significantly.

The large-scale scientific simulations generate a huge amount of data that widens the gap between the I/O capacity and computation abilities of high-end computing (HEC) machines [51]. This bottleneck for data analysis raises the need for in situ analytics where simulation data are processed prior to the I/O. Although feasible, the in situ analysis of peta-scale data incurs computation overhead in the HEC machines. The authors proposed adaptive compression service for in situ analytics middle-ware to effectively utilize available bandwidth and to optimize the performance of the HEC during end-to-end data transfer. Experimental results with gyro-kinetic simulation (GKW) on 80-node 1280 core cluster machine show that the compression ratio and available computational power are two main factors to achieve the maximum compression. The authors in [43] further proposed a framework called FlexAnalytics and profiled three compression algorithms called lossy [52], bzip2 [53], and gzip [54]. The experimental results show that all three compressions are not useful for optimized data transfer with bandwidth more than 264.19 Mb/s. This bottleneck imposes the challenge of compression/decompression time reduction to cope with the I/O needs of HEC.

Although compression-based data reduction methods are feasible for big data reduction, the processing overhead of de-compression introduces latency which lowers the performance of analytics algorithms in real-time environments. Moreover, the additional computations consume more cloud resources and increase the overall operational costs of big data systems. However, the techniques enable to store big data efficiently without significantly losing the original data in both the lossy and the lossless compression-based methods.

3.3 Data Deduplication (Redundancy Elimination)

Data redundancy is the key issue for data analysis in big data environments. Three main reasons for data redundancy are: (1) addition of nodes, (2) expansion of datasets, and (3) data replication. The addition of a single virtual machine (VM) brings around 97% more redundancy, and the growth in large datasets comes with 47% redundant data points [13]. In addition, the storage mechanism for maximum data availability (also called data replication) brings 100% redundancy at the cluster level. Therefore, effective data deduplication and redundancy elimination methods can cope with the challenge of redundancy. The workload analysis shows that the 3× higher throughput improves performance about 45% but in some extreme cases the performance degrades up to 161%. The energy overhead of deduplication is 7%; however, the overall energy saved by processing deduplicated data is 43%. The performance is degraded to 5%, whereas energy overhead is 6% for pure solid state drive (SSD) environments. However, in hybrid environment the system's performance is improved up to 17%.

Cluster deduplication is a generalized big data reduction scheme for disk-based cluster backup systems. The redundant data stored on multiple disks and partitions are a serious challenge for big data processing systems. The deduplication techniques allow to handle different data chunks (partitions) using hash functions to lower intra-node and inter-node communication overheads. In addition, these methods improve the storage efficiency by eliminating redundant data from multiple nodes. Large-scale cluster deduplication schemes face challenge of information-island (only server-level deduplication is possible due to the communication overhead) where data routing is the key issue. Another major challenge is disk-chunk-index-lookup (keeping duplicated chunk indexes of large datasets creates memory overheads), which degrades the performance of backup clients due to frequently random I/O for lookup and indexing.

Data deduplication schemes are based on either locality or similarity of data in the cluster. Locality-based approaches (stateful or stateless routing schemes) work on the location of duplicated data and perform optimization [14]. The major issue with the locality-based approach is the communication overhead of transferring similar data to same nodes. On the other hand, similarity-based schemes distribute the similar data to the same nodes across the cluster and reduce communication burden [56]. Although the schemes solve the problem of communication overhead, they prove ineffective for inter-node data deduplication system. To cope with challenges of communication overhead and ineffectiveness in inter-node deduplication systems, some hybrid techniques are also proposed in the recent literature. For example, SiLo [15] and Σ-Dedupe [12] used both the similarity- and locality-based techniques where SiLo addressed only the challenge of inter-node deduplication while Σ-Dedupe creates the balance between high deduplication and scalability across all of the nodes in the cluster. Although the cluster-level deduplication is effective for big data reduction, new deduplication methods are required to improve energy efficiency and resource awareness in large-scale data centers. The evaluation of Σ-Dedupe is performed in terms of efficiency analysis and normalized deduplication ratio using Eqs. 12 and 13. The duplication efficiency (DE) is measured using Eq. 12 as follows:

$$\mathrm{DE} = \frac{L - P}{T} = \left(1 - \frac{1}{\mathrm{DR}}\right) \times \mathrm{DT} \tag{12}$$

where physical and logical size of datasets are denoted by P and L, respectively, and T represents the processing time for deduplication. In addition, DT represents deduplication throughput and DR represents the deduplication ratio in the overall data. Similarly, the normalized effective deduplication ratio (NEDR) is used to measure the cluster-wide deduplication and storage imbalances collectively (see Eq. 13)

$$\mathrm{NEDR} = \frac{\mathrm{CDR}}{\mathrm{SDR}} \times \frac{\alpha}{\alpha + \sigma} \tag{13}$$

In Eq. 13, the CDR represents the cluster-level deduplication ratio and the SDR denotes single-node-level deduplication ratio. In addition, α represents the average usage of storage while σ shows the standard deviation of cluster-wide storage usage.

The massive amount of data movement in data centers increases the computational and communicational burdens. The exploitation of in-network data processing techniques can reduce the aforementioned complexities. The authors of [57] proposed an in-network data processing technique for bandwidth reduction by customizing routing algorithms, eliminating network redundancy (by caching frequent packets), and reducing on-path data. The proposed technique performs partial data reduction and significantly improved throughput for in-network query processing.

In contrast, mobile users in the same locality or with the same habits generate similar data points causing a huge amount of redundant data in participatory big data environments. In addition, the absence of spatiotemporal correlation among sensory data in mobile opportunistic networks is also a great challenge. The authors in [58] proposed a cooperative sensing and data forwarding framework for mobile opportunistic networks where sampling redundancy is eliminated to save energy consumption. The authors proposed two data forwarding protocols [epidemic routing with fusion and binary spray and wait with fusion (BSWF)] by leveraging data fusion. The essence of the research is the intelligent fusion of sensory data to eliminate redundancy. The simulation results revealed that proposed method can remove 93% of redundancy in the data as compared to non-cooperative methods.

The issue of data duplication or redundancy has been addressed by researchers in different environments at different levels (mobile, cluster, cloud, and data center). Therefore, the selection of best method depends upon the application models. For example, in mobile opportunistic networks and mobile crowd sensing environments, the data redundancy elimination methods are best suited when they are deployed in mobile devices. Similarly, for scientific and highly correlated data deduplication is best suitable when it is performed at cluster, data center, and cloud level.

3.4 Data Preprocessing

Data preprocessing is the second important phase of big data processing, and it must be preprocessed before storage at large-scale infrastructures [19]. This approach helps in big data reduction and also extracts the meta-data for further processing. The authors argue that primary approaches for data preprocessing are based on semantic analysis (using ontologies) or linked data structures (such as Google knowledge graph). However, this literature review uncovers few other techniques, such as low memory pre-filters for streaming data, URL filtration method, and map-reduce implementation of 2D peak detection methods in the big genomic data.

Low memory pre-filters are used for preprocessing genomic sequencing streaming data. The algorithm runs in a single pass and gives improved performance for error correction and lossy compression in data streams. In addition, the algorithm extracts the subsets of data streams using sketch-based techniques and applies pre-filtering algorithm for lossy compression and error correction. The algorithm first constructs the Bruijn graph, and the subsets are extracted using locus-specific graph analysis technique [59]. The massive data redundancy is handled using the *k*-mers median, and subsequently, digital normalization is employed as the data reduction technique. The authors argued that 95% of the data can be removed in the normal sequencing sample and the percentage reaches 98% of high-coverage single sequencing data. The results show that memory requirement for proposed algorithm is reduced from 3 TB to 300 GB of RAM.

Wearable sensors generate multi-dimensional, nonlinear, dynamic data streams with weak correlation between data points. The authors in [60] used locality-sensitive bloom filter to enhance the efficiency of instance-based learning for front-end data preprocessing near the sensing elements. The technique enables the filtration and communication of only the relevant and meaningful information to reduce computational and communication burden. The authors discussed the big healthcare data system for elderly patients and developed a prototype of the proposed solution. The architecture of the system is based upon a wearable sensor with bluetooth low energy (BLE) interface and can communicate with mobile application and/or PC to establish a personal area network (PAN). The mobile application processes the data and recognizes the state of the user. The sensor data and user states are further transmitted to a remote big data server through TCP/UDP ports. The compression algorithms are applied to incoming data streams, and resultant compressed files remain 10% of the actual data streams.

An application of big data reduction is the filtration of malicious URLs in Internet security applications. The authors in [21] proposed two feature reduction techniques that extract the lexical features and the descriptive features and then combine their results. The lexical features extract the structure of the URLs. However, the issue with lexical features is that malicious URL addresses have constantly changing behavior to abstain from malware detection software. The descriptive features are extracted to track and preserve different states of the same URL to label it as malicious. The authors selected passive-aggressive (for dense feature extraction) and confidence weighted algorithms (for sparse feature extraction) as the online learning algorithms and trained their models with extracted features [61, 62]. The prediction results of the filtration technique demonstrate around 75% data reduction with approximately 90% retention rate (inverse of data loss).

The analysis of large-scale proteomics data, which is the protein-level representation of big genomic data, requires massive computational resources to study different protein properties, such as expressions, changes in protein structures, and the interaction with other proteins. The protein molecules are too large to be identified by spectrometer and therefore are broken into smaller fragments called peptides. The mass spectrometer outputs the graphical output where each spectrum of data points is shown using Gaussian curves for peptide identification. The preprocessing step of proteomics data analysis is the identification of curves also called the 2D peaks. Each of the samples submitted to the spectrometer takes around 100 min to 4 h for complete analysis. During the passage of peptides, the spectrometer takes snapshots of spectrum every second where each peptide remains visible for several spectrums. The authors proposed a map-reduce implementations for proteomics data analysis where 2D peaks are picked at map level and further analyzed at reduce level [63]. The data reduction takes place at map level by applying preprocessing techniques for decoding the arrays, noise removal, and management of the overlapping peaks in the spectrum. Experimental results show that the given map-reduce implementation completes the data analysis in 22 min.

Recently light detection and ranging (LiDAR) technology enabled the generation of big 3D spatial data [64]. A cloud computing-based LiDAR processing system (CLiPS) processes big 3D spatial data effectively. The CLiPS uses several preprocessing techniques for data reduction to deal with large size of data. The data reduction is performed using a vertex decimation approach to provide a user's preferred parameters to reduce the big data. The results show the advantage of cloud computing technologies over the conventional systems comparing performance and time consumption.

The literature review of these techniques reveals that data preprocessing techniques are highly dependent on the nature of big data and also encourage further investigation of the underlying problem. Therefore, these techniques could not be generalized for all types of big data streams.

3.5 Dimension Reduction

Big data reduction is mainly considered to be the dimension reduction problem because the massive collection of big data streams introduces the 'curse of dimensionality' with millions of features (variables and dimensions) that increases the storage and computational complexity of big data systems [5]. A wide range of dimension reduction methods are proposed in the existing literature. The methods are based on clustering, map-reduce implementations of existing dimension reduction methods, feature selection techniques, and fuzzy logic implementations. Table 2 presents the summary of the above-mentioned methods.

The dynamic quantum clustering (DQC) enables powerful visualization of high-dimensional big data [8]. It outlines subsets of the data on the basis of density among all of the correlated variables in high-dimensional feature space. The DQC is scalable to very large systems due to its support for highly distributed data in parallel environments. The DQC is based on quantum mechanics techniques from physics. It works by constructing a potential proxy function to estimate the density of data points. The function named as parzen estimator, $\emptyset(\vec{x})$, is applied over n-dimensional feature space, and it is the sum of Gaussian functions centered at each data point, $\vec{x}$ (see Eq. 14). The DQC next defines a vector function that satisfies the Schrodinger equation (see Eq. 15). Afterward, the DQC computes Gaussians functions from subsets using Hamiltonian operator defined in the potential function and multiplies the results by quantum-time evolution operator $e^{-i\delta tH}$ (where δt is set as small, i is the ith iteration, and H is the Hamiltonian distance). The DQC then computes the new center of each Gaussian and iterates the whole procedure. The results show that large and complex datasets could be analyzed using the DQC without any prior assumption about the number of clusters or using any expert information. In addition, the DQC could be applied to noisy data to identify and eliminate unimportant features to produce better results. This data reduction strategy makes DQC useful for big data analysis.

$$\varphi(\vec{x}) = \sum_{l=1}^{m} e^{\frac{-1}{2\sigma^2}\left(\vec{x}-\overrightarrow{x_l}\right)^2} \tag{14}$$

The potential function $V(\vec{x})$ can be defined over the same n-dimensional feature space and defined as the function for which $\varphi(\vec{x})$ satisfies the time-independent Schrodinger equation.

$$\frac{-1}{2\sigma^2}\nabla^2\varphi + V(\vec{x})\varphi = E\varphi = 0 \tag{15}$$

Conventional dimensionality reduction algorithms that use Gaussian maximum likelihood estimator could not handle the datasets with over 20,000 variables. The BIG-Quic addresses the issue by applying a parallel divide-and-conquer strategy that can be applied up to 1-million variables in the feature space for dimensionality reduction [9]. The results show that the proposed algorithm is highly scalable and faster than the existing algorithms, such as Glasso and ALM [65, 66].

Knowledge discovery from high-dimensional big social image datasets is quite challenging. The authors proposed a new framework called twister which is a map-reduce implementation of k-means algorithm for dimensionality reduction [67]. The authors proposed a topology-aware pipeline-based method to accelerate broadcasting and to overcome the limitations of existing massively parallel infrastructure (MPI) implementations. In addition, the performance of the system was improved using local reduction techniques. This technique reduces local data before shuffling. The amount of data reduced is estimated using Eq. 16.

$$\text{Amount of data} = \frac{\text{No. of nodes}}{\text{No. of maps}} \times 100\% \tag{16}$$

Normally, online learning techniques take the full feature set as the input, which is quite challenging when dealing with high-dimensional features space. The authors proposed an online feature selection (OFS) approach where the online learners only work on small and fixed-length feature sets. However, the selection of active features for accurate prediction is a key issue in the approaches presented in [68]. The authors investigated sparsity regularization and truncation techniques and proposed a new algorithm called the OFS. The results showed that the OFS outperformed RAND and PE_{trun} algorithms for UCI datasets and it works best in online learning mode as compared to batch-mode learning.

The corsets are the small set of points that represent the larger population of data and maintain the actual properties of overall population. These properties vary by nature of knowledge discovery algorithms. For example, the corsets representing first k-components maps with first k-components in the big data. Similarly, the corsets containing k-clusters with radius r approximate the big data and obtain the k-clusters with same r. In this way, the authors [11] applied corsets to reduce the big data into small and

Table 2 Dimension reduction methods

References	Methods	Description	Strengths	Weaknesses
Weinstein et al. [8]	DQC	Visual data mining method	Ability to expose hidden structures and determine their significance in high-dimensional big data	Lacks efficiency Requires a combination of statistical tools
Hsieh et al. [9]	BIGQuic	Applying a parallel divide-and-conquer strategy	Supports parallelization Allowing for inexact computation of specific components	Lacks accuracy and reliability
Hoi et al. [68]	OFS	Selection of active features for accurate prediction	Works best in online learning mode as compared with batch-mode learning	Lacks efficiency
Feldman et al. [11]	Corsets	Applying corsets to reduce big data	High significance when used for data complexity	Works well on small datasets only
Azar and Hassanien [71]	LHNFCSF	Linguistic hedges fuzzy classifier	Data reduction	Lack of efficiency
Cichocki [72]	TNs	Tensor decomposition and approximation	Reduction in feature spaces	High computational complexity
Dalessandro [73]	FH	Maps features from high-dimensional space to low-dimensional spaces	Reduces feature space randomly	Compromise on data quality
Liu et al. [77]	CF	Classifier training with minimal feature spaces	Outlines critical feature dimensions and adequate sampling size	Assumptions need to be more accurate to outline critical feature dimension
Zeng and Li [74]	IPLS	Performs incremental analysis of streaming data	Computationally efficient Highly accurate	Needs to handle change detection in streaming data

manageable size, which reduces the overall data complexity. The authors mapped corsets with *k*-means, principal component analysis (PCA) and projective clustering algorithms deployed with massively parallel streaming data [69, 70]. The big data is reduced in such a way that high dimensions of input space do not affect the cardinalities of corsets. In addition, the corsets are merged by maintaining the property that union of two corsets represents the reduced set of union of two big datasets. The experimental results showed that corsets are suitable to address NP-hard problems in massively parallel and streaming data environments where big data complexity is reduced by application of data processing algorithms on small datasets that are approximated from big data. In addition, the corsets are paradigm shift in big data analysis where the focus of research remains on big data reduction instead of improving the computational efficiency of existing algorithms.

Medical big data comes across several issues regarding extraction of structures, storage of massive data streams, and uncovering the useful knowledge patterns. Research shows that fuzzy classification methods are good choice to cope with the above-mentioned issues. Recently, the authors of [71] presented linguistic hedges fuzzy classifier with selected features (LHNFCSFs) to reduce dimensions, select features, and perform classification operations. The integration of linguistic hedges in adaptive neural-fuzzy classifier enhances the accuracy. The LHNFCSF reduces the feature space effectively and enhances the performance of the classifier by removing unimportant, noisy, or redundant features. The results depict that the LHNFCSF addresses the medical big data issues by reducing the dimensions of large datasets and speeding up the learning process and improves the classification performance.

Tensors are multi-dimensional representations of data elements with at least one extra dimension as compared to matrices. The increasing numbers of elements demand more computational power to process the tensors. Tensor processing works fine with small tensors. However, processing large tensors is a challenging task [10]. Tensor decomposition (TD) schemes are used to extract small but representative tensors from large tensors [72]. Three widely used TD strategies include canonical polyadic decomposition (CPD), tucker decomposition, and tensor trains (TT). The TD schemes represent the large tensors linked with their small representations. These decomposition schemes reduce the high dimensionality in big datasets and establish the interconnection among tensors to form tensor networks (TNs). These TNs enable to further reduce the data size by using optimization-based algorithms to find factor matrices and optimize using linear and nonlinear least square methods. The case studies show that tensor decomposition strategies could be used to alleviate/

eliminate dimensionality in large scientific computing datasets and have many potential applications for feature extraction, cluster analysis, classification, data fusion, anomaly detection, pattern recognition, integration, time-series analysis, predictive modeling, multi-way component analysis, and regression.

The feature hashing (FH) method reduces feature dimensionality by randomly assigning each feature in the actual space to a new dimension in a lower-dimensional space [73]. This is done by simply hashing the ID of the original features. Usually, all dimensional reduction techniques degrade the data quality. However, most of them preserve the geometric qualities of the data. Alternately, the FH does not preserve the data quality. Research shows that the degradation of data quality is so minimal that its benefits are outweighed by the cost. The FH scales linearly with simple preprocessing and preservation of data sparsity, if exists. The scalability property of the FH makes it a natural choice for million (or even billion) feature datasets. For example, the FH method applied to email spam filtering shows its power when applied upon sparse and streaming data with real-time requirements of mass customization. The results show that the feature set is reduced from one billion to one million features.

Big data streams enter with episodic information and create high-dimensional feature spaces. Normally, the feature extraction methods need whole data in the memory that increases the computational complexity of big data systems and degrades the performance of classifiers. The incremental feature extraction methods are the best choice to handle such issues [74]. Incremental partial least squares (IPLSs) is a variant of the partial least squares method that effectively reduces dimensions from large-scale streaming data and improves the classification accuracy. The proposed algorithm works in two-stage feature extraction process. First, the IPLS adopts the target function to update the historical means and to extract the leading projection direction. Second, the IPLS calculates the rest of projection directions that are based on the equivalence between the Krylov sequence and the partial least square vectors [75]. The comparison of the IPLS was performed with incremental PCA algorithm, incremental inter-class scatter method, and incremental maximum margin criterion technique. The results revealed that the IPLS showed improved performance in terms of accuracy and computational efficiency.

Systems of systems (SoS)—case study The integration of heterogeneous and independent operating computing systems to collectively maximize the performance as compared to the individual settings leads toward the SoS [76]. Nowadays, SoS is contributing to generate big data and raises the need for data reduction. Few examples of statistical and computational intelligence tools for data reduction in SoS include the PCA, clustering, fuzzy-logic, neuro-computing, and evolutionary computing, such as genetic algorithms, and Bayesian networks. The authors applied data reduction methods at different stages of analyzing photovoltaic data that were collected from different sources. The original dataset contained 250 variables, which is highly dimensional and is not practical due to limitations of execution time and memory constraints on a desktop computer. Two approaches for data reduction at this stage were considered: (1) labeling the interesting attributes by domain expert and (2) development of an adaptive learning algorithm for automatic attribute selection. The authors employed the first approach for data reduction. The authors further cleaned-up the data and removed all invalid data points from the dataset. For example, solar irradiance in night hours generates data points with negative values, therefore not feasible for contribution in the study. After removing the invalid data, the data points containing very low values for global horizontal irradiance (GHI), direct normal irradiance (DNI), and direct horizontal irradiance (DHI) are removed to create more crispy data for further analysis.

The cleaned data are further fed into two nonparametric model generation tools, namely the fuzzy inference system generator and back-propagation neural network training tools using MATLAB fuzzy logic toolbox and the neural network toolbox. The initial evaluation of both of the tools revealed that the input variables should be further reduced for performance maximization in terms of execution time and memory consumption. The authors expanded the nonlinear data by using additional variables that in turn increased the performance of the training model but also increased time and space complexity. Therefore, the PCA is applied for dimension reduction to compress the data without significant information loss. After application of the PCA, further dimension reduction was performed using genetic algorithm (GA). First, the data were reduced using the GA on the full set of initial data and remaining data were expanded nonlinearly. Finally, the expanded dataset is used to train a neural network to assess the overall effectiveness of the GA. In practice, the time- and computation-related constraints were limited to the selection of training data to first 1000 samples. The first iteration of GA took initially 244 samples and reduced it to 74. The results showed that the GA reduced the number of attributes up to 70%.

3.6 Data Mining and Machine Learning (DM and ML)

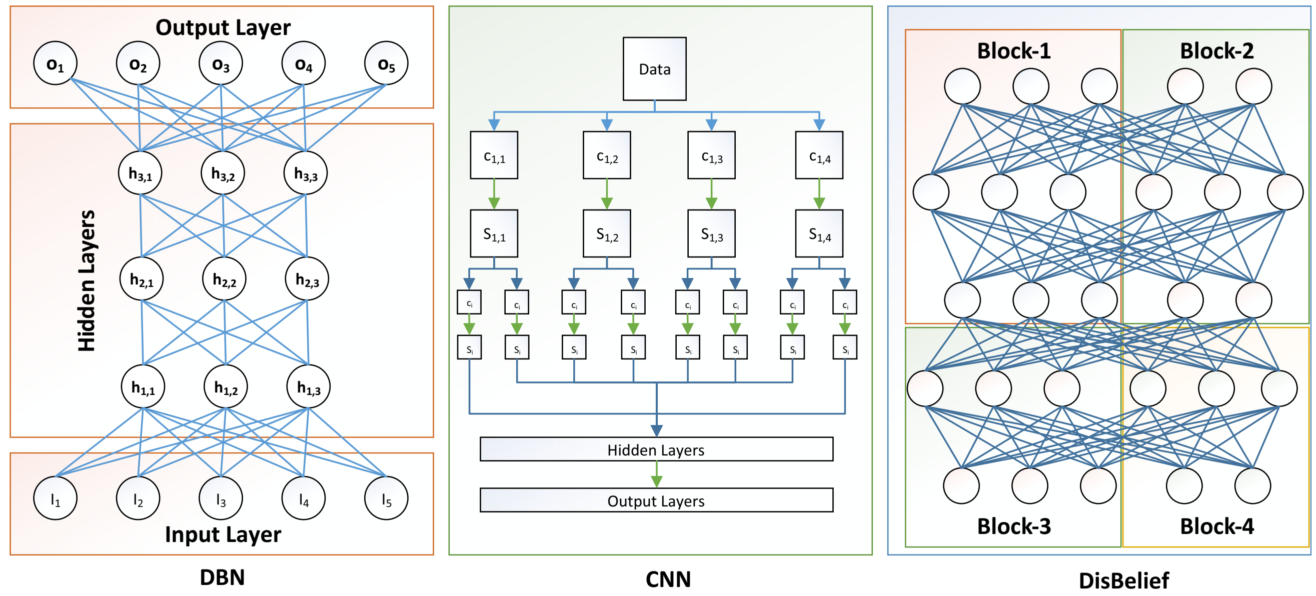
Recently, several DM and ML methods have also been proposed for big data reduction. The methods are either applied to reduce data immediately after its acquisition or customized to address some specific problems. For example, the authors [78] proposed a context-aware big data forwarding scheme using distributed wearable sensors. The scheme is based on hidden markov model (HMM) to enable context-aware features in the distributed sensor model and forwards only relevant information when there is some significant change in the data [78]. The proposed scheme reduces the communication and storage overhead in big data environment. The authors compared the result of proposed locality-sensitive hash (LSH)-based method with linear perception (LP)-based dimensionality reduction algorithm and argued on the effectiveness of the proposed scheme as compared to the LP-based dimensionality reduction methods.

The problem of mining uncertain big data due to existential probabilities becomes worse and requires huge efforts and computational power to explore the incrementally growing uncertain search space. Therefore, the search space is needed to be reduced for uncovering the maximum certain and useful patterns. The authors of [79] proposed a map-reduce algorithm to reduce the search space and mine frequent patterns from uncertain big data. The algorithm facilitates the users to confine their search space by setting some succinct anti-monotone (SAM) constraints for data analysis and subsequently mines the uncertain big data to uncover frequent patterns that satisfy the user-specified SAM constraints. The input of the algorithm is uncertain big data, user-specified minimum support (minsup) threshold, and the SAM constraints. Two sets of map-reduce implementations are used to uncover singleton and non-singleton frequent patterns. Experimental results show that the user-specified SAM (termed as selectivity) is directly proportional to multiple parameters which are derived from algorithm's runtime, the pairs returned by map function, the pairs sorted and shuffled by reduce function, and the required constraints checks.

Artificial intelligence methods, for example, artificial neural networks (ANNs) have also potential for big data reduction and compression. The authors in [80] proposed a self-organized Kohonen network-based method to reduce big hydrographic data acquired from the deep seas. The proposed system first converts the raw data into 'swath' files using a combination of four filters: (1) limit filter, (2) amplitude filter, (3) along track filter, and (4) across track filter. The appropriate combinations of the filters ensure the optimal dataset size. Despite filtering, the sample size is still large to be considered as big data. The self-organized Kohonen networks are trained and optimized using filtered hydrographic data to cluster the incoming data streams. The experimental results exhibited the feasibility of self-organized Kohonen networks for big hydrographic data for further analysis.

In addition to conventional machine learning algorithms, based on shallow learning models, deep learning is creating space as an option for big data reduction methods [81]. Deep learning models are hierarchical representations of supervised and unsupervised classification methods that are best suitable for large-scale, high-dimensional big data streams [22]. Deep learning models become computationally inefficient with the increase in big data complexity. However, the availability of MPIs (clusters/clouds) can address the aforementioned issue. Conventionally, deep learning models work at multiple layers with different granularities of information and predictive abilities. Two well-established deep learning architectures are deep belief networks (DBNs) and convolutional neural networks (CNNs). The DBN learning models are developed in two stages: (1) the initial models are developed using unsupervised learning methods with unlabeled data streams and (2) the models are fine-tuned using supervised learning methods and labeled data streams. The typical architecture of the DBN in Fig. 2 shows the multilayer representation of the deep learning model. The architecture is based on an input and output layer with multiple intermediate hidden layers. The output of each $(n-1)$th layer becomes the input of the nth layer in the architecture. In addition, the learning models are fine-tuned using back-propagation methods to support generative performance and judicial power of the DBNs. Although the CNNs are based on learning models, they differ from the DBNs. The CNNs layers are either the convolutional layers to support the convolution of several input filters or subsampling layers to reduce the size of output variable from previous layers. Effective utilization of these deep learning models in conjunction with MPIs can significantly reduce big data streams.

Deep learning models are inherently computationally complex and require many-core CPUs and large-scale computational infrastructures. Some recent learning approaches for such large-volume, complex data include locally connected networks [82, 83], improved optimizers, and deep stacking networks (DSNs). The authors of [84] proposed the hybrid deep learning model, called DisBelief, to address the issue of high dimensionality in big data streams. Disbelief utilizes a large-scale cluster to parallelize both the data and the learning models using synchronization, message passing, and multi-threading techniques. The DisBelief model first achieves parallelism by partitioning large-scale networks into small blocks that are mapped to a single node and then achieves data parallelism using two separate distribution optimization procedures called stochastic gradient descent (SGD) for online optimization and sandblaster for batch optimization. Although feasible for big data reduction, the deep learning models are resource hungry and require MPIs based on clusters of CPUs or GPUs. Therefore, there is a need to develop optimized deep learning strategies to achieve resource efficiency and reduce communication burdens inside the MPIs.



DBN CNN DisBelief

Fig. 2 Deep learning models

The wide spectrum view of the proposed methods for big data reduction uncovers the fact that the research on big data reduction methods is being carried out at several levels of big data architecture and in different forms.

4 Open Research Issues

The discussion on the open research issues, limitations, and possible future research directions is presented in this section.

Network theory The extraction of topological network and ranking of network nodes from big data is a complex process due to inherent big data complexity. In addition, the complex interactions among different nodes of the extracted networks increase the computational complexity of existing network theory-based methods. The scale-free networks and random networks can effectively reduce complex big datasets. However, the full network extraction from inconsistent and missing data is the key challenge [16, 40]. Big data systems contain many small and manageable datasets, but finding the connections among these datasets is a crucial task. The similarity graph is generated from big data where vertices represent datasets and the weighted edges are defined on the basis of similarity measure. The graph is further reduced by merging similar datasets to reduce the number of nodes. The similarity-based big data reduction methods are good choice for network extraction and reduction. However, a range of new similarity measures are required to deal with the evolving complexity and to fully comply with 6Vs of big data [17].

Persistent homology is a good solution for topological data analysis, but it involves high computational complexity. The solutions like selective collapse algorithms represent datasets in the form of forests, and the nodes are collapsed in a way to improve the speed of persistent homology and maintain strong collapse. The persistent homology tools for reducing and analyzing big data still need to be further explored in the future research [18, 42]. Similarly, the automated extraction of events and their representation in network structures is an emerging research area. The assessment of events co-occurrence and their mutual influences is the key challenge for big data reduction. The authors in [41] performed the influence assessment among different concepts (events or datasets) based on the co-occurrence of two events. The co-occurrence is assessed based on the preferential attachment property which determines that new nodes are most likely connected with highly connected nodes as compared to less connected nodes. In addition, the influence relationship among network nodes can be effectively derived from conditional dependencies among variables. However, the mathematical and probabilistic constraints increase the computational complexity in network extraction methods. Therefore, efforts are required to optimize the influence assessment methods for computationally efficient and better approximated network structures [41].

Compression Big data processing in cloud computing environments involves challenges relevant to inefficiency, parallel memory bottlenecks, and deadlocks. The spatiotemporal compression is a key solution for processing big graph data in the cloud environment. In spatiotemporal compression-based methods, the graph is partitioned and edges are mapped into different clusters where compression operations are performed for data reduction. The spatiotemporal compression is an effective approach for big data reduction. However, the research is required to find new parameters that are helpful in finding additional spatiotemporal correlations for maximum big data reduction [45].

The gap between computations and I/O capacity in the HEC systems degrades the system performance significantly. Although in situ analytics are useful for decreasing the aforementioned gap, the cost of computation increases abruptly. The compression methods can significantly reduce the transferred data and narrow the gap between computations and I/O capacity. The authors in [43] suggested that the number of available processors and the data reduction ratio (compression ratio) are two key factors that need attention in future research in this area. Alternately, the AST is a new way of compressing digitized data by selectively stretching and warping the signal. The technique is primarily based on self-adaptive stretch where more samples are associated with sharp features and fewer samples are associated with redundant coarse features. The AST performs data compression of the signal extracted on frequency domain. The method also performs inverse transformation of the constructed signal. The method specifically works with big data involving signal processing. However, the generalization to other domains is a bottleneck in this research [47].

Compressed sensing is a compressible and/or sparse signal that projects a high-dimensional data in low-dimensional space using random measurement matrix. The proposed scheme performs data acquisition and compression in parallel for improved performance as compared with Nyquist sampling theory-based compression methods. The probability of poor data quality and information fidelity loss increases when the analysis is performed on reduced and compressed data [48]. The RED encoding scheme proposed by authors in [48] is used to manage massively generated voluminous electrophysiology data. The scheme performs best when encoding of invariant signals is performed. However, while encoding time-series signals, the performance varies but the scheme achieves high compression rate with improved computational speed in lossless compression. The performance of the RED encoding methods degrades with high variance in signals [50].

Parallel compression methods can be used to reduce the data size with low computational cost. It uses proper orthogonal decomposition to compress data because it can effectively extract important features from the data and resulting compressed data can also be linearly decompressed. The parallel compression methods balance between feature retention error and compression ratio and perform fast decompression for interactive data visualization. However, the standard deviation of error is significant due to noise in the dataset [55]. The sketching method uses count-min sketch algorithm to compress vehicular movement data and achieve compact communication. Although it ensures data reduction by preserving some important characteristics of the original data, the probability of information fidelity loss is more when sketching is applied with inconsistent and noisy data stream [46].

Data deduplication (redundancy elimination) Cluster-level data deduplication is a key requirement to comply with service-level agreements (SLAs) for privacy preserving in cloud environments. The main challenge is the establishment of trade-off between high deduplication ratio and scalable deduplication throughput. The similarity-based deduplication scheme optimizes the elimination process by considering the locality and similarity of data points in both the intra-node and inter-node scenarios. The approach is effective for data reduction, but it requires to be implemented with very large-scale cluster data deduplication systems [12]. The I/O latency and extra computational overhead of cluster-level data deduplication are among the key challenges. The authors in [13] characterized the deduplication schemes in terms of energy impact and performance overhead. The authors outlined three sources of redundancy in cluster environment including: (1) the deployment of additional nodes in the cluster, (2) the expansion of big datasets, and (3) the usage of replication mechanisms. The outcomes of the analysis reveal that the local deduplication, at cluster level, can reduce the hashing overhead. However, local deduplication cannot achieve the maximum redundancy. In contrast, global deduplication can achieve maximum redundancy but compromises on the hashing overheads. In addition, fine-grained deduplication is not suitable for big datasets especially in streaming data environments [13].

Data routing is a key issue in multi-node data deduplication systems. The availability of sufficient throughput is the main bottleneck for data movement among backup and recovery systems. The stateful data routing schemes, as compared to stateless approaches, have higher overhead with low imbalance in the data which minimizes the utility of data deduplication systems. The open issues for data routing include the characterization of parameters which causes the data skew. In addition, the scalability of routing methods to large-scale cluster systems and the impact of feature selection and super-chunk size are needed to be explored in future research. Moreover, the addition of new nodes is needed to be considered for effective bin migration strategies [14].

The in-network data processing methods facilitate in data reduction and reduce the bandwidth consumption, and the efforts are required for on-the-path data reduction and redundancy elimination. The reduced bandwidth consumption by in-network data processing methods enable enhanced query processing throughput. The future implementation of in-network data processing is envisioned as the provision of network-as-a-service (NaaS) in the cloud environment which is fully orchestrated for redundancy elimination and query optimization [57]. In addition, there is a need to devise new network-aware query processing and optimization models, and integration of these models in distributed data processing systems. Research shows that co-operative sensing methods can aid in significant data reduction in large-scale sensing infrastructures [58]. Current co-operative sensing methods lack in low-level contextual features and adaptive global learning models to handle the change detection in streaming data environments. Future research work to integrate current low-level contextual models and adaptive machine learning methods can aid in maximum data reduction as well as collection of a high-quality data.

Data preprocessing The investigations of research problems relevant to preprocessing techniques of big data are still at the initial level. Most of the works are based on the adoption of existing preprocessing methods that were earlier proposed for historical large datasets and data streams. The forefront deployment of data preprocessing methods in the big data knowledge discovery process requires new, efficient, robust, scalable, and optimized preprocessing techniques for both historical and streaming big data. The application of appropriate and highly relevant preprocessing methods not only increases data quality but also improves the analytics on reduced datasets. The research on new methods for sketching, anomaly detection, noise removal, feature extraction, outliers detection, and pre-filtering of streaming data is required to reduce big data effectively. In addition, the deployment of adaptive learning models in conjunction with said methods can aid in dynamic preprocessing of big streaming data [21].

Dimension reduction Big data reduction is traditionally considered to be a dimension reduction problem where multi-million features spaces are reduced to manageable feature spaces for effective data management and analytics. Unsupervised learning methods are the key consideration for dimensionality reduction problem. However, this literature review revealed several other statistical and machine learning methods to address this issue. The techniques to combine conventional dimension reduction methods with statistical analysis methods can increase the efficiency of big data systems [8]. This approach may aid in targeting highly dense and information oriented structures (feature sets) to achieve maximum and efficient big data reduction. Alternately, tensor decomposition and approximation methods are useful to cope with the curse of dimensionality that arises due to high-dimensional complex and sparse feature spaces [10]. The main application of TD-based methods is witnessed in the scientific computing and quantum information theory domain. This literature review revealed that the issue of dimensionality reduction in big data could be handled by adopting front-end data processing, online feature selection from big data streams, constant-size corsets for clustering, statistical methods, and fuzzy classification-based soft computing approaches. These adoptions open new research avenues for interdisciplinary research and develop novel big data reduction methods. The strengths and weaknesses of these methods are already presented in detail in Table 2.

DM and ML The DM and the ML methods for big data reduction could be used at various levels of big data architectures. These methods enable to find interesting knowledge patterns from big data streams to produce highly relevant and reduced data for further analysis. For example, HMM as applied in [78] enables the context-aware features to filter the raw data streams to transmit only highly relevant and required information. In addition, the scheme enables to project high-dimensional data streams in manageable low-dimensional feature spaces. Although the application of these methods is convenient for data reduction, the trade-off between energy consumptions in local processing with raw data transmission is a key challenge that is needed to be considered. The DM and ML methods also have potential to be deployed in map-reduce implementations of Hadoop architecture. The authors in [79] parallelized the frequent pattern mining algorithms using the map-reduce programming model to reduce the massively high-dimensional feature space produced by uncertain big data. However, there exists a huge research gap for the implementation of other DM and ML methods for big data reduction that include supervised, unsupervised, semi-supervised, and hierarchical deep learning models [85]. In addition, the implementation of statistical methods, both descriptive and inferential, for big data reduction using approximation and estimation properties in uncertain big data environments is also useful for data reduction in map-reduce programming models. Moreover, the DM and ML methods are equally useful for big data reduction when coupled with artificial intelligence-based optimization methods. However, supervised, unsupervised, and semi-supervised learning methods need more attention for future research [80].

Deep learning models have recently gained attention by the researchers. The deployment of deep learning models for big data reduction is potential research direction that can be pursued in future. The deep learning models are initially developed from certain data and gradually evolve with uncertain data to effectively reduce big data streams. However, the increasing computational complexities of operating in uncertain big data environments and optimization of learning models to discover patterns from maximum data are the issues that can be further investigated [84].

In this section, we thoroughly discussed the open issues, research challenges, the limitations of proposed methods for big data reduction and presented some future research directions. The survey reveals that big data reduction is performed at many levels during the data processing lifecycle that include data capturing, data preprocessing, data indexing and storage, data analysis, and visualization. Therefore, the relevant reduction methods and systems should be designed to handle the big data complexity at all stages of big data processing. In addition, the future research work should focus on considering all 6Vs to process big data in computing systems with different form factors from fine-grained mobile computing systems to large-scale massively parallel computing infrastructures.

5 Conclusions

Big data complexity is a key issue that is needed to be mitigated. The methods discussed in this article are an effort to address the issue. The presented literature review reveals that there is no existing method that can handle the issue of big data complexity single-handedly by considering the all 6Vs of big data. The studies discussed in this article mainly focused on data reduction in terms of volume (by reducing size) and variety (by reducing number of features or dimensions). However, further efforts are required to reduce the big data streams in terms of velocity and veracity. In addition, the new methods are required to reduce big data streams at the earliest immediately after data production and its entrance into the big data systems. In general, compression-based data reduction methods are convenient for reducing volume. However, the decompression overhead needs to be considered to improve efficiency. Similarly, network theory-based methods are effective for extracting structures from unstructured data and to efficiently handle the variety in big data. The data deduplication methods are useful to improve the data consistency. Therefore, the aforementioned methods are a suitable alternative to manage the variability issues in big data. Likewise, data preprocessing, dimension reduction, data mining, and machine learning methods are useful for data reduction at different levels in big data systems. Keeping in view the outcomes of this survey, we conclude that big data reduction methods are emerging research area that needs attention by the researchers.

References

1. Wu X et al (2014) Data mining with big data. IEEE Trans Knowl Data Eng 26(1):97–107
2. Che D, Safran M, Peng Z (2013) From big data to big data mining: challenges, issues, and opportunities. In: Database systems for advanced applications
3. Battams K (2014) Stream processing for solar physics: applications and implications for big solar data. arXiv preprint arXiv:1409.8166
4. Zhai Y, Ong Y-S, Tsang IW (2014) The emerging "big dimensionality". Comput Intell Mag IEEE 9(3):14–26
5. Fan J, Han F, Liu H (2014) Challenges of big data analysis. Nat Sci Rev 1(2):293–314
6. Chandramouli B, Goldstein J, Duan S (2012) Temporal analytics on big data for web advertising. In: 2012 IEEE 28th international conference on data engineering (ICDE)
7. Ward RM et al (2013) Big data challenges and opportunities in high-throughput sequencing. Syst Biomed 1(1):29–34
8. Weinstein M et al (2013) Analyzing big data with dynamic quantum clustering. arXiv preprint arXiv:1310.2700
9. Hsieh C-J et al (2013) BIG & QUIC: sparse inverse covariance estimation for a million variables. In: Advances in neural information processing systems
10. Vervliet N et al (2014) Breaking the curse of dimensionality using decompositions of incomplete tensors: tensor-based scientific computing in big data analysis. IEEE Signal Process Mag 31(5):71–79
11. Feldman D, Schmidt M, Sohler C (2013) Turning big data into tiny data: constant-size coresets for *k*-means, pca and projective clustering. In: Proceedings of the twenty-fourth annual ACM-SIAM symposium on discrete algorithms
12. Fu Y, Jiang H, Xiao N (2012) A scalable inline cluster deduplication framework for big data protection. In: Middleware 2012. Springer, pp 354–373
13. Zhou R, Liu M, Li T (2013) Characterizing the efficiency of data deduplication for big data storage management. In: 2013 IEEE international symposium on workload characterization (IISWC)
14. Dong W et al (2011) Tradeoffs in scalable data routing for deduplication clusters. In: FAST
15. Xia W et al (2011) SiLo: a similarity-locality based near-exact deduplication scheme with low RAM overhead and high throughput. In: USENIX annual technical conference
16. Trovati M, Asimakopoulou E, Bessis N (2014) An analytical tool to map big data to networks with reduced topologies. In: 2014 international conference on intelligent networking and collaborative systems (INCoS)
17. Fang X, Zhan J, Koceja N (2013) Towards network reduction on big data. In: 2013 international conference on social computing (SocialCom)
18. Wilkerson AC, Chintakunta H, Krim H (2014) Computing persistent features in big data: a distributed dimension reduction approach. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)
19. Di Martino B et al (2014) Big data (lost) in the cloud. Int J Big Data Intell 1(1–2):3–17
20. Brown CT (2012) BIGDATA: small: DA: DCM: low-memory streaming prefilters for biological sequencing data
21. Lin M-S et al (2013) Malicious URL filtering—a big data application. In 2013 IEEE international conference on big data
22. Chen J et al (2013) Big data challenge: a data management perspective. Front Comput Sci 7(2):157–164

23. Chen X-W, Lin X (2014) Big data deep learning: challenges and perspectives. IEEE Access 2:514–525
24. Chen Z et al (2015) A survey of bitmap index compression algorithms for big data. Tsinghua Sci Technol 20(1):100–115
25. Hashem IAT et al (2015) The rise of "big data" on cloud computing: review and open research issues. Inf Syst 47:98–115
26. Gani A et al (2015) A survey on indexing techniques for big data: taxonomy and performance evaluation. In: Knowledge and information systems, pp 1–44
27. Kambatla K et al (2014) Trends in big data analytics. J Parallel Distrib Comput 74(7):2561–2573
28. Jin X et al (2015) Significance and challenges of big data research. Big Data Res 2(2):59–64
29. Li F, Nath S (2014) Scalable data summarization on big data. Distrib Parallel Databases 32(3):313–314
30. Lohr S (2014) For big-data scientists, 'janitor work' is key hurdle to insights. http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html
31. Ma C, Zhang HH, Wang X (2014) Machine learning for big data analytics in plants. Trends Plant Sci 19(12):798–808
32. Ordonez C (2013) Can we analyze big data inside a DBMS? In: Proceedings of the sixteenth international workshop on data warehousing and OLAP
33. Oliveira J, Osvaldo N et al (2014) Where chemical sensors may assist in clinical diagnosis exploring "big data". Chem Lett 43(11):1672–1679
34. Shilton K (2012) Participatory personal data: an emerging research challenge for the information sciences. J Am Soc Inform Sci Technol 63(10):1905–1915
35. Shuja J et al (2012) Energy-efficient data centers. Computing 94(12):973–994
36. Ahmad RW et al (2015) A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J Netw Comput Appl 52:11–25
37. Bonomi F et al (2014) Fog computing: a platform for internet of things and analytics. In: Big data and internet of things: a roadmap for smart environments. Springer, pp 169–186
38. Rehman MH, Liew CS, Wah TY (2014) UniMiner: towards a unified framework for data mining. In: 2014 fourth world congress on information and communication technologies (WICT)
39. Patty JW, Penn EM (2015) Analyzing big data: social choice and measurement. Polit Sci Polit 48(01):95–101
40. Trovati M (2015) Reduced topologically real-world networks: a big-data approach. Int J Distrib Syst Technol (IJDST) 6(2):13–27
41. Trovati M, Bessis N (2015) An influence assessment method based on co-occurrence for topologically reduced big data sets. In: Soft computing, pp 1–10
42. Dey TK, Fan F, Wang Y (2014) Computing topological persistence for simplicial maps. In: Proceedings of the thirtieth annual symposium on computational geometry
43. Zou H et al (2014) Flexanalytics: a flexible data analytics framework for big data applications with I/O performance improvement. Big Data Res 1:4–13
44. Ackermann K, Angus SD (2014) A resource efficient big data analysis method for the social sciences: the case of global IP activity. Procedia Comput Sci 29:2360–2369
45. Yang C et al (2014) A spatiotemporal compression based approach for efficient big data processing on Cloud. J Comput Syst Sci 80(8):1563–1583
46. Monreale A et al (2013) Privacy-preserving distributed movement data aggregation. In: Geographic information science at the heart of Europe. Springer, pp 225–245
47. Jalali B, Asghari MH (2014) The anamorphic stretch transform: putting the squeeze on "big data". Opt Photonics News 25(2):24–31
48. Wang W et al (2013) Statistical wavelet-based anomaly detection in big data with compressive sensing. EURASIP J Wirel Commun Netw 2013(1):1–6
49. He B, Li Y (2014) Big data reduction and optimization in sensor monitoring network. J Appl Math. doi:10.1155/2014/294591
50. Brinkmann BH et al (2009) Large-scale electrophysiology: acquisition, compression, encryption, and storage of big data. J Neurosci Methods 180(1):185–192
51. Zou H et al (2014) Improving I/O performance with adaptive data compression for big data applications. In: 2014 IEEE international parallel & distributed processing symposium workshops (IPDPSW)
52. Lakshminarasimhan S et al (2011) Compressing the incompressible with ISABELA: in situ reduction of spatio-temporal data. In: Euro-Par 2011 parallel processing. Springer, pp 366–379
53. Ahrens JP et al (2009) Interactive remote large-scale data visualization via prioritized multi-resolution streaming. In: Proceedings of the 2009 workshop on ultrascale visualization
54. Compression utility, gzip. http://www.gzip.org
55. Bi C et al (2013) Proper orthogonal decomposition based parallel compression for visualizing big data on the K computer. In: 2013 IEEE symposium on large-scale data analysis and visualization (LDAV)
56. Bhagwat D, Eshghi K, Mehra P (2007) Content-based document routing and index partitioning for scalable similarity-based searches in a large corpus. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining
57. Rupprecht L (2013) Exploiting in-network processing for big data management. In: Proceedings of the 2013 SIGMOD/PODS Ph.D. symposium
58. Zhao D et al (2015) COUPON: a cooperative framework for building sensing maps in mobile opportunistic networks. IEEE Trans Parallel Distrib Syst 26(2):392–402
59. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18(5):821–829
60. Cheng Y, Jiang P, Peng Y (2014) Increasing big data front end processing efficiency via locality sensitive Bloom filter for elderly healthcare. In: 2014 IEEE symposium on computational intelligence in big data (CIBD)
61. Dredze M, Crammer K, Pereira F (2008) Confidence-weighted linear classification. In: Proceedings of the 25th international conference on machine learning
62. Crammer K et al (2006) Online passive-aggressive algorithms. J Mach Learn Res 7:551–585
63. Hillman C et al (2014) Near real-time processing of proteomics data using Hadoop. Big Data 2(1):44–49
64. Sugumaran R, Burnett J, Blinkmann A (2012) Big 3d spatial data processing using cloud computing environment. In: Proceedings of the 1st ACM SIGSPATIAL international workshop on analytics for big geospatial data
65. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3):432–441
66. Scheinberg K, Ma S, Goldfarb D (2010) Sparse inverse covariance selection via alternating linearization methods. In: Advances in neural information processing systems
67. Qiu J, Zhang B (2013) Mammoth data in the cloud: clustering social images. Clouds Grids Big Data 23:231
68. Hoi SC et al (2012) Online feature selection for mining big data. In: Proceedings of the 1st international workshop on big data, streams and heterogeneous source mining: algorithms, systems, programming models and applications
69. Hartigan JA, Wong MA (1979) Algorithm AS 136: a *k*-means clustering algorithm. In: Applied statistics, pp 100–108

70. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemometr Intell Lab Syst 2(1):37–52
71. Azar AT, Hassanien AE (2014) Dimensionality reduction of medical big data using neural-fuzzy classifier. Soft Comput 19(4):1115–1127
72. Cichocki A (2014) Era of big data processing: a new approach via tensor networks and tensor decompositions. arXiv preprint arXiv:1403.2048
73. Dalessandro B (2013) Bring the noise: embracing randomness is the key to scaling up machine learning algorithms. Big Data 1(2):110–112
74. Zeng X-Q, Li G-Z (2014) Incremental partial least squares analysis of big streaming data. Pattern Recogn 47(11):3726–3735
75. Ruhe A (1984) Rational Krylov sequence methods for eigenvalue computation. Linear Algebra Appl 58:391–405
76. Tannahill BK, Jamshidi M (2014) System of systems and big data analytics–Bridging the gap. Comput Electr Eng 40(1):2–15
77. Liu Q et al (2014) Mining the big data: the critical feature dimension problem. In: 2014 IIAI 3rd international conference on advanced applied informatics (IIAIAAI)
78. Jiang P et al (2014) An intelligent information forwarder for healthcare big data systems with distributed wearable sensors. IEEE Syst J PP(99):1–9
79. Leung CK-S, MacKinnon RK, Jiang F (2014) Reducing the search space for big data mining for interesting patterns from uncertain data. In: 2014 IEEE international congress on big data (BigData congress)
80. Stateczny A, Wlodarczyk-Sielicka M (2014) Self-organizing artificial neural networks into hydrographic big data reduction process. In: Rough sets and intelligent systems paradigms. Springer, pp 335–342
81. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
82. LeCun Y et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
83. Kavukcuoglu K et al (2009) Learning invariant features through topographic filter maps. In: 2009 IEEE conference on computer vision and pattern recognition, CVPR 2009
84. Dean J et al (2012) Large scale distributed deep networks. In: Advances in neural information processing systems
85. Martens J (2010) Deep learning via Hessian-free optimization. In: Proceedings of the 27th international conference on machine learning (ICML-10), June 21–24, Haifa, Israel