Wireless Networks (2023) 29:1085–1144
https://doi.org/10.1007/s11276-022-03177-5

ORIGINAL PAPER

A comprehensive and systematic literature review on the big data management techniques in the internet of things

Arezou Naghib[1] · Nima Jafari Navimipour[2,3] · Mehdi Hosseinzadeh[4,5,6] · Arash Sharifi[1]

Accepted: 19 October 2022 / Published online: 15 November 2022

Abstract
The Internet of Things (IoT) is a communication paradigm and a collection of heterogeneous interconnected devices. It produces large-scale distributed, and diverse data called big data. Big Data Management (BDM) in IoT is used for knowledge discovery and intelligent decision-making and is one of the most significant research challenges today. There are several mechanisms and technologies for BDM in IoT. This paper aims to study the important mechanisms in this area systematically. This paper studies articles published between 2016 and August 2022. Initially, 751 articles were identified, but a paper selection process reduced the number of articles to 110 significant studies. Four categories to study BDM mechanisms in IoT include BDM processes, BDM architectures/frameworks, quality attributes, and big data analytics types. Also, this paper represents a detailed comparison of the mechanisms in each category. Finally, the development challenges and open issues of BDM in IoT are discussed. As a result, predictive analysis and classification methods are used in many articles. On the other hand, some quality attributes such as confidentiality, accessibility, and sustainability are less considered. Also, none of the articles use key-value databases for data storage. This study can help researchers develop more effective BDM in IoT methods in a complex environment.

Keyword Big data management · Internet of things · Knowledge discovery · Systematic literature review (SLR)

✉ Nima Jafari Navimipour
Navimipour@ieee.org; nima.navimipour@khas.edu.tr

Arezou Naghib
arezou.naghib@srbiau.ac.ir

Mehdi Hosseinzadeh
mehdihosseinzadeh@duytan.edu.vn

Arash Sharifi
a.sharifi@srbiau.ac.ir

[1] Present Address: Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kadir Has University, Istanbul, Turkey

[3] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

[4] Institute of Research and Development, Duy Tan University, Da Nang, Vietnam

[5] School of Medicine and Pharmacy, Duy Tan University, Da Nang, Vietnam

[6] Computer Science, University of Human Development, Sulaymaniyah 0778-6, Iraq

1 Introduction

The Internet of Things (IoT) is an emerging information technology model and a dynamic network that enables interaction between self-configuring, smart, and interconnected devices and humans [1]. The IoT's ubiquitous data collection devices (such as Radio-Frequency Identification (RFID) tags, sensors, Global Positioning Systems (GPS), Geographical Information Systems (GIS), drives, Near-Field Communication (NFC), actuators, and mobile phones) collect and share real-time, mobile, and environmental data for automatic monitoring, identification, processing, maintenance, and control in real-time [2–4]. The IoT ecosystem has five main components generally: IoT devices, including sensors and actuators that collect data and perform actions on things; IoT connectivity, including protocols and gateways, that is responsible for creating communication in the IoT ecosystem between smart devices, gateways, and the cloud; an IoT cloud that is responsible for data storage, processing, analysis, and decision-making; IoT analytics and data management are responsible for processing the data; and end-user devices and user interfaces help to control and configure the system [5]. The most important applications of IoT include environmental monitoring, disaster management, smart homes/buildings, smart farms, healthcare, smart cities, urban, smart manufacturing, intelligent transport systems, smart floods, financial risk management, supply chain management, water management, enterprise culture, cultural heritage, smart surveillance, military tracking and environment, digital forensics, underwater environments, and understanding social phenomena [6–22]. The IoT devices and sensors in the Wireless Sensor Networks (WSN) generate large data. According to the international data corporation[1] forecast, the number of IoT devices will be 41.6 billion and generate 79.4 zettabytes of data in 2025. This massive structured, semi-structured, and unstructured data, which is expanding rapidly with time, results in "Big Data" [23]. "Big data" technologies are a new generation of distributed architectures and technologies that provide distributed data mining capabilities to inexpensively, valuable, and effectively extract value from a huge dataset with characteristics such as volume, velocity, variety, variability, veracity, and value [24]. Big data provides both opportunities and problems for organizations and enterprises. Big data can improve data precision, be used for forecasting and decision-making, and give stakeholders more in-depth analytical findings [2]. Traditional data processing systems cannot collect, process, manage, and interpret data effectively using conventional mechanisms. Therefore, it requires a scalable architecture or framework for effective capture, storage, management, and analysis [25].

A major challenge in implementing IoT in real and complex environments is analyzing heterogeneous data volumes that contain a wide variety of knowledge content [26]. Various platforms, tools, and technologies have been developed for big data monitoring, collecting, ingesting, storing, processing, analysis, and visualization [10, 27]. These platforms and tools are Apache Hadoop, MapReduce, 1010data, Apache Storm, Cloudera, Cassandra, HP-HAVEn, SAP-Hana, Hortonworks, MongoDB, Apache Kafka, Apache Spark, Infobright, etc. Industries and enterprises use Big Data Analytics (BDA) with IoT technologies to handle the timely analysis of information streams and intelligent decision-making [28–30]. BDM in the IoT involves different analytic types [31]. Marjani et al. [29] discussed analytical types in real-time, offline, memory, business intelligence, and at massive levels. Singh and Yassine [28] divided analytical types into preprocessing, pattern mining, and classification. Gandomi and Haider [32] divided big data processing into two major phases: data management and data analytics. Also, Ahmed et al. [33] provided five aspects of big data: acquisition and storage; programming model; benchmark process; analysis; and application. Finally, ur Rehman et al. [34] divided BDA into five main steps: data ingestion, cleaning, conformation, transformation, and shaping.

However, despite the importance of BDM in the IoT and the rising challenges in this area, as far as we know, there is not any complete and detailed systematic review in this field. Hence, this paper tries to analyze the mechanisms of BDM in the IoT. The main contributions of this paper are as follows:

- Presenting a study of the existing methods for BDM in the IoT.
- Dividing BDM methods in the IoT are divided into four main categories: BDM processes, BDM architectures/frameworks, quality attributes, and big data analytics types.
- Dividing the BDM process in the IoT into six main steps, including data collection, communication, data ingestion, data storage, processing and analysis, and post-processing.
- Dividing the BDM architecture/framework in the IoT into two main subcategories: BDM architectures/frameworks in IoT-based applications and BDM architectures/frameworks in the IoT paradigms.
- Exploring the primary challenges, issues, and future works for BDM in the IoT.

The following subsection discusses related work to show the main differences between this review and similar studies. Also, the abbreviations used in this paper are presented in Table 1.

1.1 Related work and contributions of this review

This section studies some reviews and survey articles that work on BDM in the IoT to highlight the need for reviewing them. In addition, this section describes the main advantages and disadvantages of this article to distinguish this one.

Ahmed et al. [27] analyzed several techniques for IoT-based big data. This article categorizes the literature based on parameters, including big data sources, system components, big data enabling technologies, functional elements, and analytics types. The authors also discussed connectivity, storage, quality of services, real-time analytics, and benchmarking as the critical requirements for big data processing and analytics.

Constante Nicolalde et al. [35] overviewed the technical tools used to process big data and discussed the relationship between BDA and IoT. The big data challenges are

[1] https://www.idc.com/.

Table 1 Paper abbreviations

Abbreviation	Definition	Abbreviation	Definition
ABC	Artificial Bee Colony	LST	Least Slack Time algorithm
ADASYN	Adaptive Synthetic Sampling	LSTM	Long Short-Term Memory
AI	Artificial Intelligence	LTE	Long Term Evolution
AMQP	Advanced Message Queuing Protocol	LWLR	Locally Weighted Linear Regression
ANN	Artificial Neural Network (ANN)	MDL	Minimum Description Length
BCN	Bayesian Convolution Network	ML	Machine Learning (ML)
BDA	Big Data Analytics	MLP	Multi-Layer Perceptron
BDM	Big Data Management	MQTT	Message Queuing Telemetry Transport
CART	Classification and Regression Tree	MIoT	Multiple Internet of Things
CDMA	Code-division multiple access	*NB*	Naive Bayes
CDNN	Convolutional Deep Neural Networks	NFC	Near-Field Communication
CEP	Complex Event Processing	NRDD-DBSCAN	New Resilient Distributed Datasets Density-Based Spatial Clustering
CFS	Correlation Feature selection	OCSTuM	One-Class Support Tucker Machine
CoAP	Constrained Application Protocol	OSSO	Swallow Swarm Optimization
DBSCAN	Density-based spatial clustering	PCA	Principal Component Analysis
DE	Differential Encoding	PPSO	Parallel Particle Swarm Optimization
DFS	Distributed File Systems	PSO	Particle Swarm Optimization
EBS	Electronic Batchload Service	QoE	Quality of Experience
EC2	Amazon Elastic Compute Cloud	QoS	Quality of service
EDL	Enhanced Deep Learning	REPtree	Reduced-Error Pruning Tree
EHO	Elephant Herd Optimization	RFID	Radio-Frequency Identification
EPL	Event Processing Language (EPL)	RNN	Recurrent Neural Network
ETL	Extract, Transform, and Load	RPL	Routing Protocol for Low-Power and Lossy Networks
GA	Genetic Algorithm	SLR	Systematic Literature Review
GIS	Geographical Information Systems	SIoT	Social Internet of Things
GM	*Grey* Prediction *Model*	SSL	Secure Sockets Layer
GPS	Global Positioning Systems	SVD	Singular Value Decomposition
HDFS	*Hadoop Distributed File System*	SVM	Support Vector Machine
HPC	High Performance Computing (HPC)	UCI	University of California, Irvine
IoT	Internet of Things	*WIT120*	Wise Information Technology of 120
KNN	K-nearest Neighbors Algorithms	WSN	Wireless Sensor Networks

divided into four general categories: data storage and analysis; the discovery of knowledge and computational complexities; information security; and scalability and data visualization.

Talebkhah et al. [36] investigated the architecture, challenges, and opportunities of big data systems in smart cities. This article suggested a 4-layer architecture for BDM in smart cities. The layers of this architecture are data acquisition, data preprocessing, data storage, and data analytics. This article also considered the opportunities and challenges for smart cities, such as heterogeneity, design and maintenance costs, failure management, throughout, etc.

Bansal et al. [37] investigated state-of-the-art research on IoT and BDM. This article proposed a taxonomy based on BDM in the IoT applications, including smart transport, smart cities, smart buildings, and smart living. BDM steps are considered as data acquisition, communication, storage, processing, and retrieval. Also, the related surveys on BDM were divided into three general categories: surveys on IoT BDA, domain-specific surveys on IoT big data, and surveys on challenges in IoT big data. The authors classified the articles based on four major vendor services (Google, Amazon, Microsoft, and IBM) to integrate IoT and IoT big data with case studies. The big data management challenges in the IoT are considered based on 13 V's challenges.

Marjani et al. [29] investigated state-of-the-art research efforts directed toward big IoT data analytics and proposed a new architecture for big IoT data analytics. This article discusses big IoT data analytic types under real-time, off-line, memory-level, business intelligence, and massive level analytics categories.

Simmhan and Perera [38] presented the analytics requirements of IoT applications. They defined the relationship between data volume capacity and processing latency of new big data platforms. This article divided decision systems into visual analytics, alerts and warnings, reactive systems, control and optimization, complex systems, knowledge-driven intelligent systems, and behavioral and probabilistic systems.

Shoumy et al. [39] discussed frameworks and techniques for multimodal big data analytics. They divided multimodal big data analytics techniques into four topics: affective framework; multimodal framework; big data and analytics framework; and fusion techniques. Furthermore, Ge et al. [40] discussed the similarities and differences among big data technologies used in IoT domains and developed a conceptual framework. This article interpreted big data research and application opportunities in eight IoT domains (healthcare, energy, transportation, building automation, smart cities, agriculture, industry, and military) and discussed the advantages and disadvantages of big data technologies. In addition, it examined four aspects of big data processes: storage, cleaning/cleansing, analysis/analytics, and visualization.

Siow et al. [41] considered the analytics infrastructure from data generation, collection, integration, storage, and computing. This article presented a comprehensive classification of analytical capabilities consisting of five categories: descriptive, diagnostic, discovery, predictive, and prescriptive analytics. In addition, a 3-layered taxonomy of data analytics was presented, including data, analytics, and applications.

Fawzy et al. [42] investigated the techniques and technologies of IoT systems from BDA architectures and software engineering perspectives. This article proposed a taxonomy based on BDA systems in the IoT, including smart environments, human, network, energy, and environmental analytics. The BDA target, approach, technology, challenges, software architecture and design, model-driven engineering, separation of concerns, and system validation and verification. The authors presented the IoT data features as multidimensional, massive, timely, heterogeneous, inconsistent, traded, valuable, and spatially correlated. The proposed domain-independent BDA-based IoT architecture has six layers. The layers of architecture are data manager, system resources controller, system recovery manager, BDA handler, software engineering handler, and security manager.

Zhong et al. [43] investigated using BDA and data mining techniques in the IoT. This article divided the review articles into four categories: architecture and platform, framework, applications, and security. The data mining methods for BDA in the IoT were discussed in these four categories. The challenges investigated in the article are as follows: data volume, data diversity, speed, data value, security, data visualization, knowledge extraction, and real-time analysis.

Hajjaji et al. [44] discussed applications, tools, technologies, architectures, current developments, challenges, and opportunities in big data and IoT-based applications in smart environments. This article divided the benefits of combining the IoT and big data into six categories: multi-source and heterogeneous data; connectivity; data storage; data analysis; and cost-effectiveness.

Ahmadova et al. [45] discussed big data applications in the IoT. They proposed a taxonomy of big data in the IoT that includes healthcare, smart cities, security, big data algorithms, industry, and general view. In the article, the authors discussed big data technologies' advantages and disadvantages for IoT domains. Also, the evaluation factors that are considered in the article are security, throughput, cost, energy consumption, reliability, response time, and availability.

Table 2 shows the summary contributions of related survey articles. The publication year, methodology, discussion, and other disadvantages are shown for each article in this table. Due to the existing weaknesses in the review articles, this paper presents a systematic literature review and a proper categorization of BDM mechanisms in the IoT that addresses the shortcomings as follows:

1. This paper provides a complete research methodology that includes research questions and the article selection process.
2. This paper discusses the newly proposed mechanisms for BDM in the IoT between 2016 and August 2022.
3. This paper considers the architectures/frameworks of IoT-based applications, including healthcare, smart cities, smart homes/buildings, intelligent transport, traffic control and energy, urban planning, and other IoT applications (smart IoT systems, smart flood, smart farms, disaster management, laundry, digital manufacturing, and smart factory).
4. This paper investigates the quality attributes and categorizes the review articles based on the quality attributes used and the reference model of standard software quality attributes, i.e., ISO 25010.
5. This paper classifies the review articles based on BDA types in the IoT and their tactics.
6. This paper considers the big data storage systems and tools in the IoT based on relational databases, NoSQL

Table 2 Comparison of the related survey articles

Reference	Year	Methodology	Discussion							Other disadvantages
			Article selection process	Newly proposed mechanisms	Analytic types' tactics	BDM process in IoT	Relevant datasets	Quality attributes	Challenges and open issues	
Ahmed et al. [27]	2017	Review	×	×	×	Partially	×	✔	✔	-Consideration of only some IoT applications -Does not storage systems and tools
Constante Nicolalde et al. [35]	2018	Survey	×	×	×	✔	×	×	✔	-There are no categories based on some factors
Talebkhah et al. [36]	2021	Survey	✔	✔	×	✔	×	×	✔	-There are no categories based on some factors -Investigation only seven use cases related to smart cities
Bansal et al. [37]	2020	SLR	×	✔	×	✔	×	×	✔	-Does not investigate big data processing and analytics tools -Consideration of only four IoT applications
Marjani et al. [29]	2017	Survey	×	×	✔	✔	×	×	✔	-Provides a general view -Focuses only on data mining strategies for BDA methods -There are no categories in this article based on some factors -Presents some BDA types, methods, and technologies for big Data mining
Simmhan and Perera [38]	2016	Review	✔	×	×	✔	×	×	✔	-Classifies the big data platforms only based on latency and throughput -Only investigates real-time big data platforms
Shoumy et al. [39]	2020	Survey	✔	✔	×	✔	×	×	✔	-Does not consider big data analytics frameworks and tools
Ge et al. [40]	2018	Survey	✔	×	Partially	✔	×	×	×	–
Siow et al. [41]	2018	Survey	✔	×	✔	✔	×	×	✔	-Does not consider important aspects of big data processing
Fawzy et al. [42]	2022	SLR	✔	✔	×	✔	×	✔	✔	-Consideration of only some IoT applications
Zhong et al. [43]	2022	Systematic survey	✔	✔	✔	✔	×	Partially	✔	-Does not investigate storage systems and tools -Consideration of only some IoT applications

Table 2 (continued)

Reference	Year	Methodology	Discussion							Other disadvantages
			Article selection process	Newly proposed mechanisms	Analytic types' tactics	BDM process in IoT	Relevant datasets	Quality attributes	Challenges and open issues	
Ahmadova et al. [45]	2021	SLR	✓	✓	×	✓	×	Partially	Partially	-Does not investigate storage systems and tools -Consideration of only some IoT applications -Does not investigate big data processing and analytics tools -Consideration of only some quality attributes
Hajjaji et al. [44]	2021	Systematic review	✓	✓	×	✓	×	Partially	✓	-Consideration of only some quality attributes -Consideration of only some IoT applications

databases, distributed file systems, and cloud/edge/fog/mist storage.

7. This paper discusses the BDM process in six steps: data collection, communication, data ingestion, data storage, processing and analysis, and post-processing, and proposes the main tools in each step.
8. This paper presents open issues and challenges on BDM in the IoT and divides challenges into two categories: BDM in the IoT and quality attributes challenges.

The rest of the paper is structured as follows: Sect. 2 explains the research methodology and the article selection process. The categories of the BDM methods in the IoT and their comparison are described in Sect. 3. Section 4 discusses the challenges and some open issues. Finally, Sect. 5 represents the conclusion and the paper's limitations.

2 Research methodology

Systematic literature review (SLR) is a research methodology that examines data and findings of the researchers relative to specified questions [46, 47]. It aims to find as much relevant research on the defined questions as possible and to use explicit methods to identify what can reliably be said based on these studies [48, 49]. This section provides an SLR to understand the BDM techniques in the IoT. The following subsection will explain the research questions and the article selection process.

2.1 Research questions

This study focuses more explicitly on the articles related to BDM in the IoT, focusing on their advantages and disadvantages, architectures, processing and analysis methods, storage systems, evaluation metrics, and tools. To achieve the goals mentioned above, the following research questions are presented.

RQ1: What is BDM in IoT?

Section 1 answered this question.

RQ2: What is the importance of BDM in the IoT?

This question aims to show the number of published articles about BDM in IoT between 2016 and August 2022.

Section 2 answers this question.

RQ3: How are the articles searched and chosen to be assessed?

Section 2.2 discusses the question.

RQ4: What are the classifications of BDM methods in the IoT?

This question aims to show the existing methods of BDM in the IoT environment. Section 3 will discuss this answer.

RQ5: What are the challenges and technical issues of BDM in the IoT?

This question identifies the challenges for BDM in the IoT and provides open issues for future research. Section 4 will discuss this answer.

2.2 Article selection process

In this study, the article's search and selection process consists of three stages. These stages are shown in Fig. 1. In the first stage, the articles between 2016 and August 2022 were searched based on the keywords and terms (presented in Table 3). These articles are the results of searching popular electronic databases. These electronic databases include Google Scholar, Elsevier, ACM, IEEE Explore, Emerald Insight, MDPI, Springer Link, Taylor and Francis, Wiley, JST, Dblp, DOAJ, and ProQuest. The articles include journals, chapters, conference papers, books, notes, technical reports, and special issues. 751 articles were found in Stage 1. In Stage 2, there are two steps to select the final number of articles to review. First, the articles are considered based on the inclusion criteria in Fig. 2. There are 314 articles left at this stage. Next, the review articles are removed; of the remaining 314 articles in the previous stage, 85 (27.07%) were review articles. Elsevier has the highest number of review articles (31.76%, 27 articles). EMERALD and Taylor and Francis have the lowest number of reviewed articles (2.35%, one article). The highest number of published review articles is in 2019 (24.71%), and the lowest is in 2022 (8.24%). The number of remaining articles at this stage is 229. In Stage 3, the title and abstract of the articles are reviewed. Also, to ensure that the articles are relevant to the study, we reviewed the methodology, evaluation, discussion, and conclusion sections. The number of selected articles retained at this stage is 110. Elsevier publishes most of the selected articles (30.91%, 34 articles). The lowest number is related to ACM (0.91%, one article). 2018 has the highest number of published articles (26.36%, 29 articles). The Future Generation Computer Systems journal publishes the highest number of articles (11.82%, 13 articles).

3 Big data management approaches in the IoT

This section presents four different categories for the reviewed articles. These categories include the BDM process in the IoT (Sect. 3.1), BDM architectures/frameworks for IoT applications (Sect. 3.2), quality attributes (Sect. 3.3), and big data analytics types (Sect. 3.4). Each category has subcategories that will be considered in its relevant section. Figure 3 shows this taxonomy.

3.1 Big data management process in the IoT

This section categorizes articles based on BDM process mechanisms and presents a comprehensive framework for BDM in the IoT. The comprehensive framework for BDM in the IoT is shown in Fig. 4. The steps of BDM in IoT include data collection, communication, data ingestion, storage, processing and analysis, and post-processing.

3.1.1 Data collection

A variety of sources generates IoT data. There are different mechanisms for IoT data collection, but there is still no fully efficient and adaptive mechanism for IoT data collection [50]. This paper divides IoT sources into sensors, applications, devices, and other resources. Figure 5 shows the classification of the sources based on these four categories.

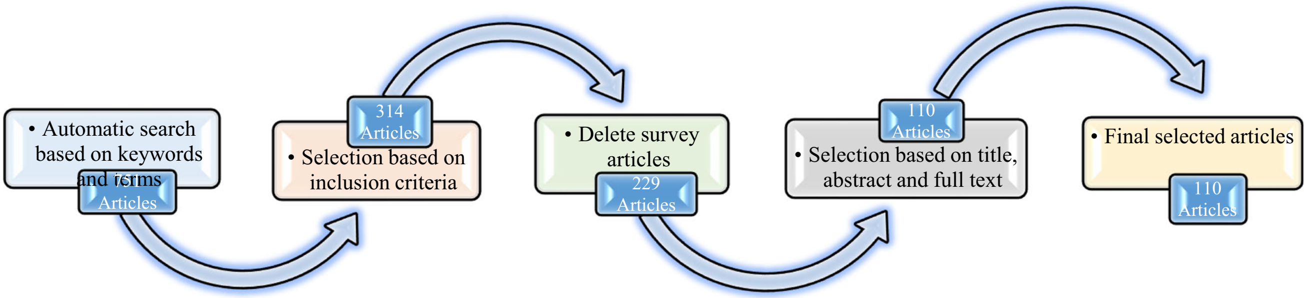

• Automatic search based on keywords and terms
751 Articles
314 Articles
• Selection based on inclusion criteria
• Delete survey articles
229 Articles
110 Articles
• Selection based on title, abstract and full text
• Final selected articles
110 Articles

Fig. 1 Articles search and selection process stage

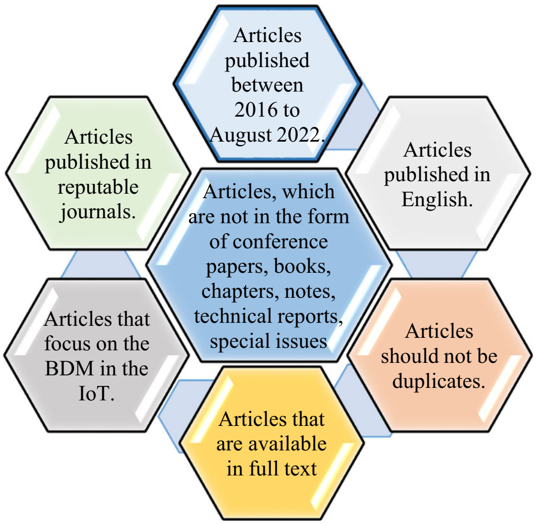

Articles published between 2016 to August 2022.
Articles published in reputable journals.
Articles published in English.
Articles, which are not in the form of conference papers, books, chapters, notes, technical reports, special issues
Articles that focus on the BDM in the IoT.
Articles should not be duplicates.
Articles that are available in full text

Fig. 2 Inclusion criteria in the articles selection process

3.1.2 Communication

The data sources are located on various networks, such as IoT sensor networks, wired and wireless sensor networks, fiber-optic sensor networks, and machine-to-machine communications. Communication technologies are required to process and analyze these data sources [51, 52]. There are several communication technologies and protocols in the IoT. The communication protocols used in the articles are IPV6, RPL, MQTT, CoAP, SSL, AMQP, Websocket, 6LowPANIPV6, Alljoyn, TCP/IP, HTTP/IP. Communication technologies are compared based on frequency, data rate, range, power usage, cost, latency, etc. There are several categories of these communication technologies. This paper divides big data communication technologies in the IoT based on distance criteria into three categories: pan, local, and WAN. Table 4 shows the articles' classification based on these three categories. Wi-Fi, ZigBee, Bluetooth, and 4G LTE are of the utmost importance in communication technology, with a total number of 29, 19, 17, and 17 articles, respectively.

3.1.3 Data ingestion

Data ingestion is the process of importing and transporting data in different formats from various sources (shown in Fig. 4) to a storage medium, processing and analyzing platform, and decision support engines [93, 94]. The quality of the dataset used by ML-based prediction models (classification) plays a vital role in BDM in the IoT. A prediction model requires a lot of correctly labeled data for correct construction, assessment, and accurate result generation [95]. Therefore, the data ingestion layer should handle the enormous volume, high speed (velocity), variety, value, variable, and validated data for the processing and analysis step. In different articles, this layer has multiple tasks. The data ingestion layer in [96] includes identification, filtration, validation, noise reduction, integration, transformation, and compression. The data ingestion layer in [97] provides data synchronization, data slicing, data splitting, and data indexing. Also, the data ingestion layer in [98] includes data stream acquisition, data stream extraction, enrichment, integration, and data stream distribution. Finally, the data ingestion layer in [99] includes data cleaning, data integration, and data compression.

There are three categories of data ingestion technologies: real-time data ingestion, batch data ingestion, and both. Real-time data ingestion is used for time-sensitive data and real-time intelligent decision-making. Batch data ingestion is used for data collection from sources at regular intervals (daily reports and schedules) [100]. There are many tools and platforms for data ingestion, such as Apache Kafka, Apache NIFI, Apache Storm, Apache Flume, Apache Sqoop, Apache Samza, Apache Minifi, Confluent Platform, and Elastic Logstash. These tools can be compared based on throughput, latency, scalability, and security [98]. The data ingestion layer in this paper includes data cleaning, data integration, data transformation/ discretization, and data reduction. Each of these steps uses special tools, methods, and algorithms. Table 5 shows the categorization of articles based on the tools that are

Table 3 Search keywords and terms

S#	Search keywords and terms
S1	"IoT big data" or "Internet of things big data"
S2	"data management in IoT" or "data management in the Internet of things"
S3	"big IoT data mining" or "big Internet of things data mining"
S4	"data mining in IoT" or "data mining in the Internet of things"
S5	"big data in IoT" or "big data in the Internet of things"
S6	"big data analytics in IoT" or "big data analytics in the Internet of things"
S7	"IoT data analyze" or "Internet of things data analyze"

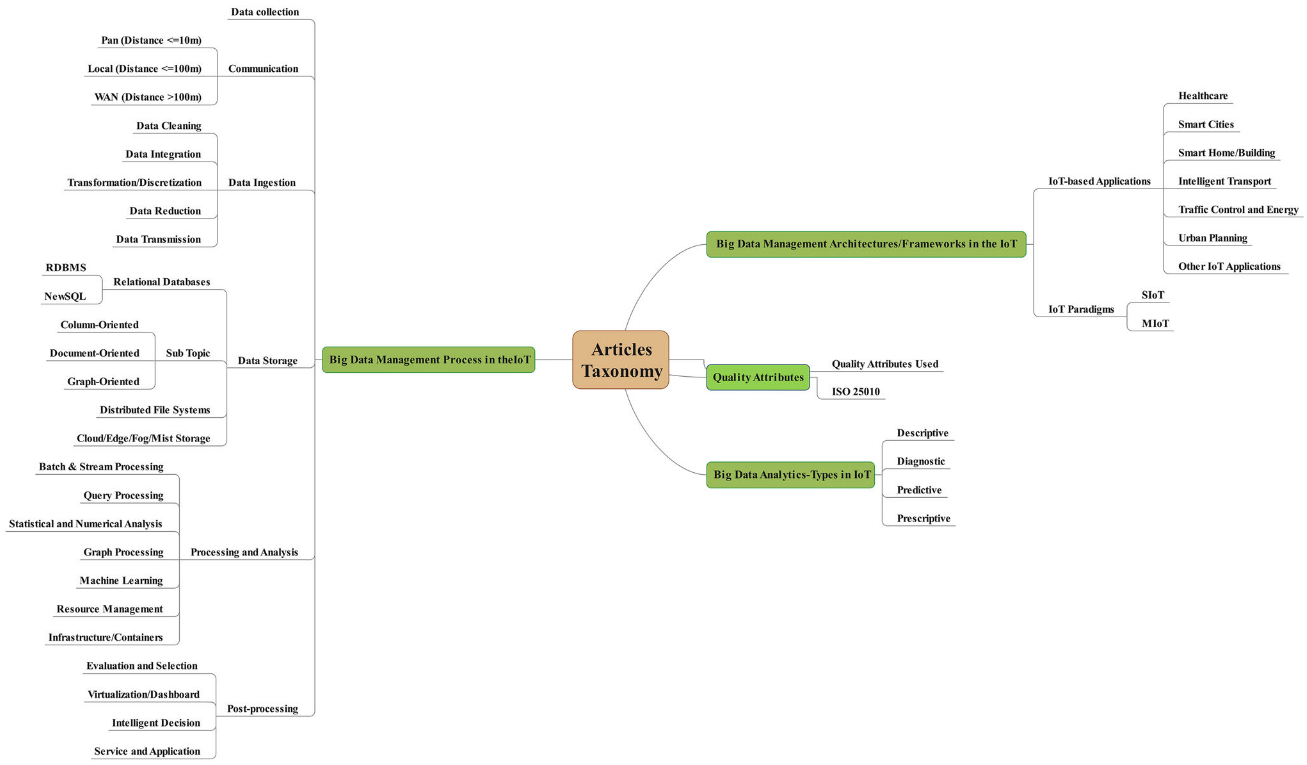


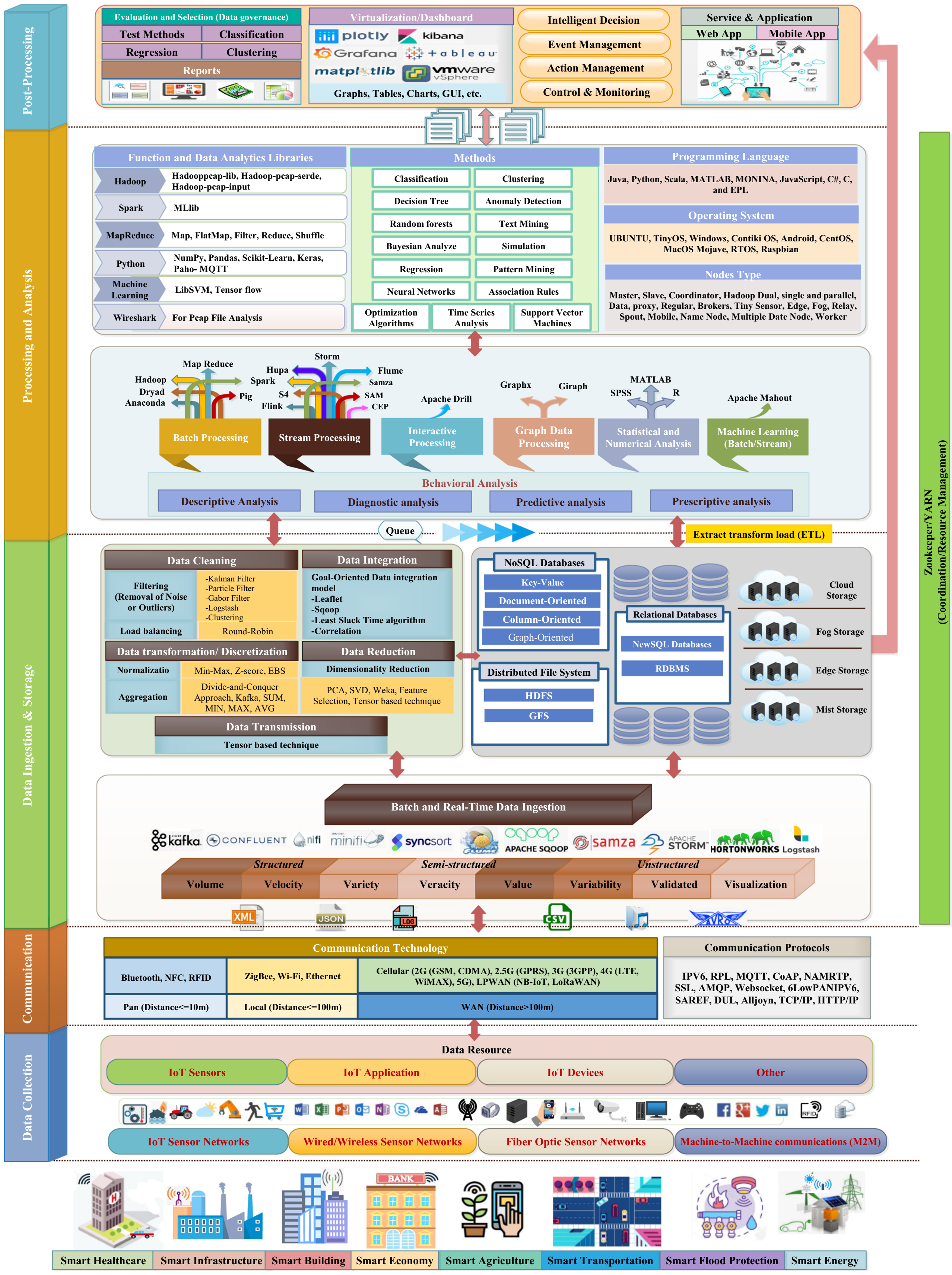
Fig. 3 Taxonomy of the selected articles

used for data ingestion. Data ingestion tools have been compared based on ingestion type, throughput, reliability, latency, scalability, security, and fault tolerance. Platforms in some articles use a combination of these tools, such as the Horton data flow platform in [101], including Apache NiFi/MiNiFi, Apache Kafka, Apache Storm, and Druid tools. As you can see in Table 5, Apache Kafka is of utmost importance to the data ingestion tool, with a total of 8 articles. Also, Table 6 shows the categorization of articles based on the big data preprocessing stage in the IoT.

3.1.4 Data storage

This subsection categorizes articles based on storage mechanisms. The articles use various methods and tools to store big data. This study divides these mechanisms into four categories: relational, NoSQL, Distributed File Systems (DFS), and cloud/edge/fog/mist storage. Each of these categories has subcategories. One of the most critical big data challenges is the categorization and scalability that traditional relational databases such as MySQL, SQL Server, and Postgres cannot overcome. Therefore, NoSQL databases are used to store big data. NoSQL technologies are divided into four categories: key-value, column-oriented, document-oriented, and graph-oriented [102]. These NoSQL technologies have many platforms to support their operations. Key-value storage is the most straightforward and highly flexible type of NoSQL database and stores all the data as a pair of keys and values. A document-oriented database stores data as a set of columns. In a relational database, data is stored in rows and read row-by-row. A graph database focuses on the relationships between data elements, and each element is stored as a node. Tables 7 and 8 show the types of storage methods used in articles. Table 7 shows the classification of articles based on relational databases, NoSQL databases, and DFS. As you can see, any of the 110 selected articles do not use the key-value databases. In relational databases, Hive, NoSQL databases, Hbase, and distributed file systems, HDFS is most commonly used. Table 7 compares these storage tools and platforms based on in-memory database/storage or disk-based, data type, scalability, security, availability, flexibility, performance, fault-tolerant, easy to use, and replication.

Table 8 shows the classification of articles based on cloud/edge/fog/mist storage. Cloud computing provides scalable computing, high data storage, processing power, and ensures the quality of the applications. However, it has main challenges such as latency, network overhead, bandwidth, data privacy, lower real-time responsiveness, location awareness, security, reliability, data availability, and accessibility [103]. Network architectures came into


Post-Processing
Processing and Analysis
Data Ingestion & Storage
Communication
Data Collection
Zookeeper/YARN
(Coordination/Resource Management)
Evaluation and Selection (Data governance)
Test Methods
Classification
Regression
Clustering
Reports
Virtualization/Dashboard
plotly
kibana
Grafana
tableau
matplotlib
vmware vSphere
Graphs, Tables, Charts, GUI, etc.
Intelligent Decision
Event Management
Action Management
Control & Monitoring
Service & Application
Web App
Mobile App
Function and Data Analytics Libraries
Hadoop
Hadooppcap-lib, Hadoop-pcap-serde, Hadoop-pcap-input
Spark
MLlib
MapReduce
Map, FlatMap, Filter, Reduce, Shuffle
Python
NumPy, Pandas, Scikit-Learn, Keras, Paho- MQTT
Machine Learning
LibSVM, Tensor flow
Wireshark
For Pcap File Analysis
Methods
Classification
Clustering
Decision Tree
Anomaly Detection
Random forests
Text Mining
Bayesian Analyze
Simulation
Regression
Pattern Mining
Neural Networks
Association Rules
Optimization Algorithms
Time Series Analysis
Support Vector Machines
Programming Language
Java, Python, Scala, MATLAB, MONINA, JavaScript, C#, C, and EPL
Operating System
UBUNTU, TinyOS, Windows, Contiki OS, Android, CentOS, MacOS Mojave, RTOS, Raspbian
Nodes Type
Master, Slave, Coordinator, Hadoop Dual, single and parallel, Data, proxy, Regular, Brokers, Tiny Sensor, Edge, Fog, Relay, Spout, Mobile, Name Node, Multiple Date Node, Worker
Map Reduce
Hadoop
Dryad
Anaconda
Pig
Hupa
Spark
S4
Flink
Storm
Flume
Samza
SAM
CEP
Apache Drill
Graphx
Giraph
MATLAB
SPSS
R
Apache Mahout
Batch Processing
Stream Processing
Interactive Processing
Graph Data Processing
Statistical and Numerical Analysis
Machine Learning (Batch/Stream)
Behavioral Analysis
Descriptive Analysis
Diagnostic analysis
Predictive analysis
Prescriptive analysis
Queue
Extract transform load (ETL)
Data Cleaning
Filtering (Removal of Noise or Outliers)
-Kalman Filter
-Particle Filter
-Gabor Filter
-Logstash
-Clustering
Load balancing
Round-Robin
Data Integration
Goal-Oriented Data integration model
-Leaflet
-Sqoop
-Least Slack Time algorithm
-Correlation
Data transformation/ Discretization
Normalizatio
Min-Max, Z-score, EBS
Aggregation
Divide-and-Conquer Approach, Kafka, SUM, MIN, MAX, AVG
Data Reduction
Dimensionality Reduction
PCA, SVD, Weka, Feature Selection, Tensor based technique
Data Transmission
Tensor based technique
NoSQL Databases
Key-Value
Document-Oriented
Column-Oriented
Graph-Oriented
Distributed File System
HDFS
GFS
Relational Databases
NewSQL Databases
RDBMS
Cloud Storage
Fog Storage
Edge Storage
Mist Storage
Batch and Real-Time Data Ingestion
kafka
CONFLUENT
nifi
minifi
syncsort
FLUME
APACHE SQOOP
samza
APACHE STORM
HORTONWORKS
Logstash
Structured
Semi-structured
Unstructured
Volume
Velocity
Variety
Veracity
Value
Variability
Validated
Visualization
XML
JSON
LOG
CSV
AVRO
Communication Technology
Bluetooth, NFC, RFID
ZigBee, Wi-Fi, Ethernet
Cellular (2G (GSM, CDMA), 2.5G (GPRS), 3G (3GPP), 4G (LTE, WiMAX), 5G), LPWAN (NB-IoT, LoRaWAN)
Pan (Distance<=10m)
Local (Distance<=100m)
WAN (Distance>100m)
Communication Protocols
IPV6, RPL, MQTT, CoAP, NAMRTP, SSL, AMQP, Websocket, 6LowPANIPV6, SAREF, DUL, Alljoyn, TCP/IP, HTTP/IP
Data Resource
IoT Sensors
IoT Application
IoT Devices
Other
IoT Sensor Networks
Wired/Wireless Sensor Networks
Fiber Optic Sensor Networks
Machine-to-Machine communications (M2M)
BANK
Smart Healthcare
Smart Infrastructure
Smart Building
Smart Economy
Smart Agriculture
Smart Transportation
Smart Flood Protection
Smart Energy

◀**Fig. 4** Big data management framework in IoT

existence to overcome these challenges, such as fog, edge, and mist computing, that move the data and computation closer to the consumer and reduce some of the workloads from the cloud [104].

Fog computing is a type of decentralized computing that is between cloud storage and IoT devices. Fog computing reduces service latency, bandwidth, energy consumption, storage, and computing costs and improves the QoS [149]. The fog computing for the IoT model supports real-time services, mobility, and geographic distribution [150]. Another alternative approach to cloud computing is edge computing. Data storage and processing in edge computing occur closer to the device or data source to improve data locality, performance, and decision-making [151]. Edge computing is less scalable than fog computing but provides near real-time analytics and high-speed data access and reduces data leakage during transmission [104, 152]. Mist computing is an intermediate layer between fog/cloud and edge computing. It can improve the fog/cloud challenges, such as response time, location awareness, data privacy, local decision-making, network overhead, latency, and computing and storage costs. Mist nodes had low processing power and storage [153]. In some articles, in addition to using cloud/edge/fog/mist storage, HDFS and NoSQL databases are used alongside these technologies.

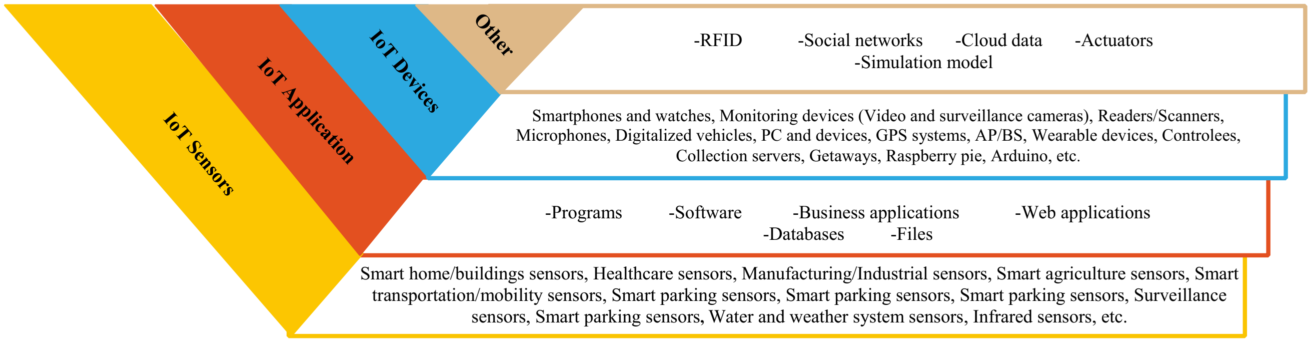

Other
-RFID
-Social networks
-Cloud data
-Actuators
-Simulation model
IoT Devices
Smartphones and watches, Monitoring devices (Video and surveillance cameras), Readers/Scanners, Microphones, Digitalized vehicles, PC and devices, GPS systems, AP/BS, Wearable devices, Controlees, Collection servers, Getaways, Raspberry pie, Arduino, etc.
IoT Application
-Programs
-Software
-Business applications
-Web applications
-Databases
-Files
IoT Sensors
Smart home/buildings sensors, Healthcare sensors, Manufacturing/Industrial sensors, Smart agriculture sensors, Smart transportation/mobility sensors, Smart parking sensors, Smart parking sensors, Smart parking sensors, Surveillance sensors, Smart parking sensors, Water and weather system sensors, Infrared sensors, etc.

Fig. 5 Big data sources categories in IoT

Table 4 Classification of articles based on communication technologies

Category Name	Communication Technologies			The Article's Ref#	#Articles
Pan (distance < = 10 m)	Bluetooth			[13, 52–67]	17
	NFC			[53]	1
	RFIID			[3, 53, 68–71]	6
Local (distance < = 100 m)	ZigBee			[51–53, 55, 56, 58, 59, 61–64, 71–78]	19
	Wi-Fi			[11, 30, 51, 52, 55–58, 60–63, 65–67, 71, 73–75, 79–88]	29
	Ethernet			[3, 51, 53, 65, 66, 74]	6
WAN (distance > 100 m)	Cellular	2G	GSM	[52]	1
			CDMA	[3, 53]	2
			NA	[8]	1
		2.5G	GPRS	[3, 53, 66]	3
		3G	3GPP, etc	[8, 51–53, 55, 58, 60, 61, 63, 66, 73–75, 79, 84, 89]	16
		4G	LTE	[8, 51–53, 55, 60, 61, 63, 66, 73–75, 79, 85, 86, 90, 91]	17
			WiMAX	[51, 53, 66, 72, 74, 85]	6
			NA	[57, 65, 76, 84, 89]	5
		5G		[8, 53, 65, 72, 73, 75, 90]	7
	LPWAN	Narrowband-IoT		[73, 75]	2
		LoRaWAN		[53, 56, 59, 92]	4

Table 5 Classification of articles based on big data ingestion tools in IoT

Data Ingestion Tools and Platforms	Features									The Article's Ref#
	Platform	Batch	Stream	Throughput	Reliability	Latency	Scalability (Horizontally)	Security	Fault-Tolerant	
Apache Kafka	Java, Scala	✓	✓	Very High	Very High	Very Low	Very High	High	✓	[59, 83, 101, 105–109]
Apache NIFI	Java	×	✓	High	High	Low	High	High	✓	[101]
Apache Storm	Java, Clojure	✓	✓	High	High	Very Low	Medium	High	✓	[52]
Apache Flume	Java	×	✓	High	High	Low	High	High	✓	[53, 107, 110]
Apache Sqoop	Java	Static	×	High	NA	High	Limited	High	✓	[53, 106, 107, 111]
Apache Samza	Java, Scala	✓	✓	Very High	High	Very Low	High	×	✓	[70]
Apache Minifi	Java	×	✓	High	High	Low	High	High	✓	[101]
Confluent Platform	Java, Scala	×	✓	High	Very High	Low	Very High	High	✓	[105]
Lambda Architecture	NA	✓	✓	Low	High	Low	High	NA	✓	[112]
Elastic Logstash	Java, JRuby	×	✓	High	High	Low	High	✓	×	[113]

The goal is to overcome the disadvantages of these technologies by using them together.

3.1.5 Processing and analysis

Big data processing and analysis in the IoT are techniques or programming models for extracting knowledge from large amounts of data for supporting and providing intelligent decisions [154]. Efficient big data processing and analysis in IoT can help mitigate many challenges in event management, action management, control and monitoring, improved customer service, cost savings, improve business relationships [155], etc. This paper divides the big data processing and analysis step in IoT into a set of sub-steps: batch and stream processing, query processing, statistical and numerical analysis, graph processing, ML, resource management, and infrastructure/containers. Table 9 shows the articles' classification and comparison of the tools based on criteria: throughput, reliability, availability, latency, scalability, security, flexibility, ease of use, and cost-effectiveness. Big data processing in the IoT is generally done at both batch and stream levels. Many tools, platforms, and frameworks exist for batch and stream processing. The tools used in the articles are Apache Hadoop, Apache Spark, Map Reduce, Apache Storm, Apache Flink, Anaconda, Apache S4, Weka, streaming analytics manager, and CEP.

As you can see in Table 9, Apache Hadoop, MapReduce, and Apache Spark are the most critical quality attributes, with a total number of 45, 32, and 31 articles, respectively. Some of these tools include a set of libraries and procedures for efficient processing and analysis. In the study, the libraries and functions used by the articles are Hadooppcap-lib, Hadoop-pcap-serde, Hadoop-pcap-input (Apache Hadoop), MLlib, GraphX, Spark Streaming, Spark SQL, Spark Core (Apache Spark), Map, FlatMap, Filter, Reduce, Shuffle (Map Reduce), Gelly, FlinkML, Table and SQL, FlinkCEP (Apache Flink), NumPy [132], Keras [108], Pandas [59], and Scikit-Learn, Paho-MQTT (Anaconda). Also, various algorithms and methods are used to process and analyze data, such as classification, clustering, regression, optimization algorithms, and SVM. Most of these tools have these algorithms.

3.1.6 Post-processing

The post-processing step is another vital task in knowledge discovery from big data in the IoT. This paper divides the post-processing step into evaluation and selection (data governance), virtualization/dashboard, intelligent decision, and service and application. The evaluation and selection stage evaluates results obtained using test methods on different types of datasets. There are various criteria for

Table 6 Classification of articles based on big data preprocessing stage in IoT

Data Preprocessing Stage	Tasks	Methods			
Data cleaning	Filtering	DBSCAN-based outlier detection [83]	Average value or a linearly interpolated value [59]		Python regular expressions (RegEx) [56]
	(Removal of noise or outliers and missing data handling)	NRDD-DBSCAN [114]	K-mediods clustering algorithm [119]		By setting thresholds [62]
		CFS and IG [115]	Particle Filter (PF) [69]		Semantic web-based filtering [121]
		HBase [63]	Gabor filter [64]		OCSTuM [122]
		Range checking and ambiguity checking [63]	Attribute mean restricted [61]		KSQL [105]
		Logstash [113]	Classification [51]		Renyi entropy [123]
		Kalman filter [30, 52, 89, 92, 116]	Savitzky-Golay filter [120]		ADASYN [124]
		Accommodative Bloom Filter [117]			Collaborative filtering [3]
		High-noise feature filter [118]			
		Association rules [79]			
		General data cleaning/filtering methods [12, 14, 53, 55, 57, 58, 65, 66, 70, 72, 74, 78, 80, 84, 86, 107, 108, 111, 125–133]			
	Load balancing	Round robin [92]	General load balancing methods [52, 57, 58, 61, 66, 74, 89, 129]		
Data integration	-	Goal-oriented data integration model [134]	Correlation [86]		General data integration methods [11, 56, 63, 78, 89, 105, 107]
		LST [92]	Python open libraries [59]		
Data reduction	-	PCA [69, 78, 114, 118, 135, 136]		SVD [75]	
		Sqoop [107]		Linear discriminant analysis [78, 127]	
		Leaflet [113]		BigReduce [137]	
		Numerosity reduction [79]		Huffman and DE techniques [65]	
				Hadoop MapReduce [120]	
		Feature selection	Pearson correlation analysis [59]	EHO [64]	Fuzzy rough [84]
			Wrapper feature selection [115]	ABC [52]	Divide and Conquer PCA [78]
			ML-based [111]	CFS [115]	GA [122]
			OSSO [124]	Quantum EHO [138]	Fuzzy rules and valuable features selections [133]
			High-variance feature removal [118]		
		General data reduction methods [56, 83, 89, 139]			
Data transformation/ discretization	Normalization	Min–Max [30, 53, 55, 63, 65, 73, 84, 85, 89, 92, 130, 131, 136]	EBS [80]		General normalization methods [59, 90, 108, 128]
		Z-score [52, 53]	Sample standard deviation [124]		
			ETL [80]		
	Aggregation	Divide-and-conquer approach [89, 92]	Spark Streaming [53]		
		Apache Kafka [101]	General aggregation methods [12, 30, 51, 52, 56–59, 61, 63, 66, 69, 70, 72, 74, 78, 79, 84, 88, 107, 108, 114, 128, 129, 140]		
		SUM, MIN, MAX, AVG [121]			
		KSQL [105]			

Table 7 Classification of articles based on relational databases, NoSQL databases, and DFS

Database/ Storage Type	Type	Name	Features						
			In-memory Database/ Storage	Data Type			Scalability	Written in	Security
				Structured	Unstructured	semi-structured			
Relational	RDBMS	SQL	✔	✔	✔	–	Vertically	C and C + +	High
		Apache HIVE	×	✔	–	–	Horizontally	Java	High
		MySQL	✔	✔	–	–	Vertically (Horizontal: Possible)	C and C + +	High
		Spark SQL	✔	✔	–	✔	Horizontally	Scala	High
		PostgreSQL	✔	✔	–	✔	Horizontally	C	High
	NewSQL Database	VoltDB	✔	✔	–	–	Horizontally	Java	Low
NoSQL Database	Column-Oriented	Apache HBase	✔	✔	–	✔	Horizontally	Java	Low
		Apache Kudu	×	✔	–	–	Horizontally	C + +	Low
		Apache Cassandra	✔	✔	–	✔	Horizontally/Vertically	Java	Low
		Apache Parquet	×	✔	–	–	Horizontally	Java	Low
	Document–Oriented	MongoDB	✔	–	✔	–	Horizontally	C + +	Low
		Elasticsearch	✔	✔	✔	–	Horizontally	Java	Low
	Graph Oriented	Neo4j	✔	✔	✔	–	Horizontally	Java	Low
		FlockDB	✔	✔	✔	–	Horizontally	Scala, Java, Ruby	Low
Distributed File Systems		HDFS (Database file: HBase)	×	✔	✔	✔	Horizontally/Vertically	Java	Low
		Google file system	×	✔	–	–	Horizontally	C/C + +	Low

Database/ Storage Type	Type	Name	Features						The article's ref#
			Availability	Flexibility	Performance	Fault-tolerant	Easy to use	Replication	
Relational	RDBMS	SQL	✔	✔	✔	✔	✔	✔	[51, 52, 72]
		Apache HIVE	✔	✔	✔	✔	✔	✔	[51, 52, 55, 89, 92, 101, 106, 111, 116]
		MySQL	✔	✔	✔	✔	✔	✔	[140]
		Spark SQL	✔	✔	✔	✔	✔	✔	[53, 59, 106, 124]
		PostgreSQL	✔	✔	✔	✔	✔	✔	[111]
	NewSQL Database	VoltDB	✔	✔	✔	✔	✔	✔	[51, 52]

Table 7 (continued)

Database/ Storage Type	Type	Name	Features						The article's ref#
			Availability	Flexibility	Performance	Fault-tolerant	Easy to use	Replication	
NoSQL Database	Column-Oriented	Apache HBase	✓	✓	✓	✓	✓	✓	[51, 52, 55, 63, 72, 80, 89, 92, 101, 107, 116, 128, 141, 142]
		Apache Kudu	✓	✓	✓	✓	✓	✓	[107]
		Apache Cassandra	✓	✓	✓	✓	✓	✓	[59, 84, 88, 105, 142]
		Apache Parquet	–	✓	✓	–	✓	–	[59]
	Document–Oriented	MongoDB	✓	✓	✓	✓	✓	✓	[54, 58, 83, 92, 108, 109, 111, 116, 125, 126, 134, 142]
		Elasticsearch	✓	✓	✓	✓	✓	✓	[105, 110, 113]
	Graph Oriented	Neo4j	✓	✓	✓	✓	✓	✓	[92]
		FlockDB	✓	✓	✓	✓	✓	✓	[92]
Distributed File Systems		HDFS (Database file: HBase)	✓	✓	✓	✓	✓	✓	[6, 7, 30, 51–53, 55, 57–59, 61–63, 66, 74, 80, 89, 92, 101, 106–108, 110, 112, 116, 118, 126, 128, 130, 143–145]
		Google file system	✓	✓	✓	✓	✓	✓	[143]

assessing the results. In this section, the articles are categorized based on the methods they used for the test. These methods are divided into four categories, including test methods, classification, clustering, and regression. Each of them uses various criteria for evaluation. Table 10 shows the articles' classification based on these four categories. The virtualization/dashboard stage uses tools, graphs, tables [75], graphical user interface [59], and charts [92] to display the results. Intelligent decisions can be made using stochastic binary decisions [156], ML, pattern recognition, soft computing, and decision models [51, 53, 74]. These tools are Kibana, Plotly, Tableau, Microsoft Power BI, Grafana, vSphere, NodeJS, and Matplotlib [59, 105, 106, 109, 110, 113, 140].

Tables 11 and 12 show the relevant datasets that the articles used for investigating/numerically assessing techniques for BDM in the IoT. These datasets are divided into two categories: 1) categorized based on characteristics including dataset name, repository, dataset characteristics, attribute characteristics, number of instances/size, and number of attributes 2) categorized based on characteristics including dataset name, website address, and size. As you can see, the UCI machine learning repository has been repeatedly used in articles as a repository to access techniques for BDM in the IoT.

3.2 Big data management architectures/frameworks in the IoT

This subsection investigates and analyzes the articles that (71 articles) presented the frameworks and architectures for BDM techniques in the IoT. These articles are divided into two categories: BDM architectures/frameworks in the IoT-based applications (63 articles) and BDM architectures/frameworks in the IoT paradigms (8 articles).

3.2.1 Big data management architectures/frameworks in the IoT applications

The architectural models used in the selected articles are layered, component-based, and cloud/fog-based architecture. A layered architecture is organized hierarchically, and each layer performs a service. The layered architecture ensures the system is more adaptable to emerging technologies at each layer and improves the acquisition and integration of data processes [167]. Component-based architecture is a framework that decomposes the system into reusable and logical components. The advantages of component-based architecture are increased quality, reliability, component reusability, and reduced time. Operations and components related to processing or storage in cloud-based or fog-based architectures are placed in the cloud or fog. Most of the proposed architectures are layered, and the

Table 8 Classification of articles based on the cloud/edge/fog/mist storage

The Article's Ref#	Cloud	Fog	Edge	Mist	HDFS	Ceph	Cassandra	MongoDB	Hbase	Hive	Tools/Platforms
[11, 14, 60, 68, 78, 81, 82, 85, 87, 90, 129, 137, 140, 146, 147]	✓	–	–	–	–	–	–	–	–	–	Cloudlet [81], Cloud Confederation [146]
[54]	✓		✓	–	–	–	–	✓	–	–	Nimbus
[56, 71, 75, 86]	✓	✓	–	–	–	–	–	–	–	–	Azure [75], Amazon EC2 [71, 75]
[141]	✓	✓	–	–	–	–	–	–	✓	–	Amazon EC2, Amazon s3, Google Cloud, IBM Cloud, AWS Cloud, Amazon EMR
[7, 30, 59]	✓	✓	–	–	✓	–	–	–	–	–	Openstack [59]
[62, 112, 118, 143, 144]	✓	–	–	–	✓		–	–	–	–	Openstack [112], Cloudlet [143]
[73]	✓	–	✓	–		✓	–	–	–	–	Openstack
[65, 121, 148]	✓	–	✓	–		–	–	–	–	–	–
[84]	✓	✓		✓		–	✓	–	–	–	–
[101]	✓	✓	✓	–	✓	–	–	–	✓	✓	–

most common types of BDM architectures in the IoT are 3-layer and 4-layer (22 and 20 articles). Also, most of the proposed architectures are in IoT-based healthcare, equivalent to 33.33%, followed by IoT-based smart cities, which equals 22.22%. The selected articles in this study used nine different OS for BDM in the IoT. Ubuntu is the most important OS, with 18 articles. Articles used programming languages to analyze and process big data in the IoT. Java, Python, and MATLAB are the major programming languages. In the following, these architectures and frameworks will be examined. For a better presentation, we have divided these architectures and frameworks into seven categories in terms of IoT applications (healthcare, smart cities, smart home/building, intelligent transport, traffic control and energy, urban planning, and other IoT applications (smart IoT systems, smart flood, smart farms, disaster management, laundry, digital manufacturing, and smart factory)). Then we review the attributes of the architectures and frameworks, including layers, the functions of the layers, the operating system, the programming language, and the advantages and disadvantages of each.

3.2.1.1 BDM architectural/framework for IoT-based healthcare Predicting health and disease and preventing deaths are essential in our modern world [168, 169]. Healthcare IoT (e.g., electronic and mobile health) uses wireless body sensor networks for monitoring the patients' environmental, physiological, and behavioral parameters [170]. Wearables and other IoT devices within the healthcare industry generate a large amount of data. The health data must be collected, stored, processed, and analyzed for future intelligent decision-making. BDA plays a vital role in minimizing computation time, predicting the future status of individuals, providing reliable health services, prevention, healthy living, population health, early detection, and optimal management [133, 158, 171]. There are the BDM mechanisms' objectives and requirements for different types of medical data [172]. Various research has presented many mechanisms for BDM in IoT-based healthcare that have advantages and disadvantages. Therefore, this subsection examines the articles (21 articles; 33.33%) that discussed the architectures or frameworks of BDM in IoT-based healthcare.

Rathore et al. [58] proposed Hadoop-based intelligent healthcare using a BDA approach. This system collected the big data and directed them to a 3-unit smart building for storing and processing. The units of this system are big data collection, Hadoop processing, and analysis and decision. This system used the 5-layer architecture for parallel, real-time, and offline processing. The layers of this architecture are the data collection, communication,

Table 9 Classification of articles based on IoT big data processing and analysis

Processing and Analysis Sub Process	Tools and Platform	Features							
		Witten in	Open-source	Batch	Stream	Throughput	Reliability	Latency	Scalability
Batch & Stream Processing	Apache Hadoop	Java	✓	✓	×	High	✓	High	High
	Apache Spark	Scala	✓	✓	✓	High	✓	Medium	High
	MapReduce	Java	✓	✓	×	High	✓	High	High
	Apache Storm	Java, Clojure	✓	-	✓	High	✓	Very Low	Medium
	Apache Flink	Java, Scala	✓	✓	✓	Very High	✓	Very Low	High
	Anaconda	Python	✓	✓	×	High	NA	Low	✓
	Apache S4	Java	✓	×	✓	Low	✓	High	High
	Weka	Java	✓	✓	×	High	✓	Low	High
	SAM	NA	✓	×	✓	High	✓	Low	High
	CEP	NA	✓	×	✓	High	✓	Very Low	High
Query Processing	Apache Pig	Java	✓	✓	✓	NA	✓	High	Limited
Statistical and Numerical Analysis	SPSS	Java	✓	✓	✓	NA	✓	NA	✓
	R	C	✓	✓	×	High	NA	Low	✓
Graph Processing	Spark Graphx	Scala	✓	×	×	NA	✓	NA	Medium
	Apache Giraph	Java	✓	×	×	NA	✓	Low	High
Machine Learning	Apache Mahout	Java, Scala	✓	✓	×	NA	✓	Low	✓
Resource Management	Zookeeper	Java	✓	✓	✓	High	✓	Low	High
	Yarn	NA	✓	✓	✓	High	✓	Low	✓
Infrastructure/Containers	Docker	Go	✓	✓	✓	–	✓	Low	✓

Processing and Analysis Sub Process	Tools and Platform	Features					The Articles's Ref#
		Security	Easy to use	Availability	Flexibility	Cost-effective	
Batch & Stream Processing	Apache Hadoop	✓	×	✓	✓	✓	[6, 7, 12, 30, 52, 52, 53, 55, 57, 58, 60, 61, 63, 64, 72–74, 74, 76, 78, 80, 89, 91, 92, 105–108, 108, 111, 112, 118–120, 120, 124, 126, 128, 129, 135, 143–145, 145]
	Apache Spark	✓	✓	✓	✓	✓	[6, 51–53, 57, 59, 61, 63, 84, 107, 112, 116, 126, 128, 129] [30, 60, 70, 74, 84, 91, 92, 106, 108–110, 114, 118, 124, 129, 135]
	MapReduce	✓	×	✓	✓	✓	[6, 30, 51–53, 55, 57, 58, 61–64, 66, 72–74, 89, 91, 111, 116, 119, 120, 124, 128–130, 135, 136, 139, 141, 143, 145]
	Apache Storm	✓	✓	✓	✓	✓	[12, 51, 52, 60, 70, 80, 83, 101, 107, 126, 128]
	Apache Flink	✓	✓	✓	✓	✓	[70, 126]
	Anaconda	✓	✓	NA	✓	✓	[107]
	Apache S4	✓	NA	NA	✓	NA	[51]

Table 9 (continued)

Processing and Analysis Sub Process	Tools and Platform	Features					The Articles's Ref#
		Security	Easy to use	Availability	Flexibility	Cost-effective	
	Weka	✔	✔	✔	✔	✔	[52, 75, 76, 79, 83, 84, 115, 132]
	SAM	NA	NA	✔	✔	NA	[101]
	CEP	✔	NA	✔	✔	✔	[59]
Query Processing	Apache Pig	NA	✔	✔	✔	✔	[101, 141]
Statistical and Numerical Analysis	SPSS	✔	✔	✔	✔	×	[12]
	R	✔	×	NA	✔	✔	[111]
Graph Processing	Spark Graphx	NA	✔	✔	✔	✔	[53, 57, 60, 66, 74, 89, 92, 124, 128, 129]
	Apache Giraph	NA	✔	✔	✔	✔	[74]
Machine Learning	Apache Mahout	✔	✔	✔	✔	✔	[58, 60, 76, 108, 111]
Resource Management	Zookeeper	✔	✔	✔	✔	✔	[54, 70, 105]
	Yarn	✔	✔	✔	✔	✔	[6, 73, 89, 107]
Infrastructure/Containers	Docker	✔	✔	✔	✔	✔	[105, 108, 112]

Table 10 Classification of articles based on evaluation and selection methods

Evaluation and Selection Methods	Type and metrics		
Test methods	Significance test [123]	Friedman test [69, 145]	FIESTA-IoT testbed [125]
	Post-Hoc test [69]	Levene's test [156]	Wilcoxon rank [145]
	Kruskal–Wallis test [69]	IoT-LAB platform [77]	
Classification	Recall or sensitivity [3, 9, 30, 64, 71, 83, 90, 111, 115–117, 120–122, 124, 126, 130, 136, 139, 141, 158]		Kappa [115]
	Precision [6, 71, 76, 83, 111, 115–117, 120, 130, 139, 141, 156, 158]		Matthews correlation coefficient [83, 115]
	Accuracy [6, 30, 52, 64, 69, 71, 73, 76, 83–85, 87, 90, 109, 111, 115–117, 120, 122, 124, 126, 130, 133, 136, 139, 141, 148, 158]		Operating characteristic [83, 115]
	Specificity [30, 64, 83, 90, 111, 115, 117, 120, 124, 126, 130, 136, 139]		Pearson product-moment correlation coefficient [111, 131]
			F1-Score [30, 71, 115, 130, 139, 141, 145, 158]
			Confusion matrix [6, 90, 158]
Clustering	Silhouette coefficient [56, 106, 135, 159]	Calinski-Harabasz index [135]	Adjusted mutual information [114]
	Normalized mutual information [119, 135]	Adjusted rand index [114, 160, 161]	E* [160, 161]
	V-measure [114, 145]	Completeness [114]	Fowlkes-Mallows score [135]
	Homogeneity [114]		
Regression	Mean absolute error [75, 108, 141]	R-squared [73, 131, 141]	The total sum of squared error [141]
	Mean square error [6, 75, 108, 141]	Q-squared [141]	U-statistics [75]
	Mean absolute percentage error [9, 75, 131, 141]	Sum of squared error [141]	Average relative errors [62, 73]
	Root mean square error [9, 59, 69, 108, 141]	Mean absolute deviation [53]	Mean square deviation [141]
	Normalized root mean square [71]		

Table 11 Datasets details

Dataset name	Repository	Dataset characteristics	Attribute characteristics	Number of instances	Number of attributes	Ref#
Diabetes data set	UCI	Multivariate, Time series	Categorical, Integer	13,437	20	[58, 61]
ICU data set	UCI	Multivariate, Time series	Real	7931	NA	[58, 61]
WISDM WISDM-raw file	WISDM Lab	Time Series Data	Real	1,098,207	6	[58, 61]
WISMDM-Transformed file		Time Series Data	Real	5424	46	[58, 61]
Drug dataset with side effects	SIDER 4.1	NA	NA	111,079	NA	[140]
Diseases with drugs details	Med Expert	NA	NA	881	NA	[140]
Cleveland heart disease database	UCI	Multivariate	Categorical, Integer, Real	303	76	[133, 141]
Indian liver patient	UCI	Multivariate	Categorical, Integer, Real	583	10	[61]
Heart rate data	NA	Multivariate, Time series	–	416	54	[61]
Multiple intelligent monitoring in intensive care	Physio Net	Multivariate, Time series	Real	72	4	[106]
Healthcare dataset	UCI	NA	NA	19,908	NA	[30]
COVID-19 data set	Hospitals of Khyber-Pakhtunkhwa Pakistan	NA	Categorical, Integer, Real	26,000	22	[158]
Arrhythmia data set	UCI	Multivariate	Multivariate	452	279	[109]
Heart disease	UCI	Multivariate	Categorical, Integer, Real	282	38	[139]
Liver disease	UCI	Multivariate	Categorical, Integer, Real	180	12	[139]
Kidney disease	UCI	Multivariate	Real	320	17	[139]
Lung disorders	UCI	Multivariate	Integer	44	42	[139]
Hepatitis	UCI	Multivariate	Categorical, Integer, Real	155	19	[52, 64, 124, 138]
Statlog (Heart) Data Set	UCI	Multivariate	Categorical, Real	270	13	[123]
Breast Cancer Data Set	UCI	Multivariate	Categorical	286	9	[3]
Gas sensors for home activity monitoring	UCI	Multivariate, Time series	Real	919,438	11	[52, 84, 115]
Gas Sensor Array Drift Dataset	UCI	Multivariate	Real	13,910	128	[135]
Electric power consumption data set	UCI	Multivariate, Time series	Real	2,075,259	9	[115]
Air quality	UCI	Multivariate, Time series	Real	9,358	15	[52]
GPS trajectories	UCI	Multivariate	Real	163	15	[52, 64, 120, 124, 138]
Indoor user movement prediction from RSS	UCI	Multivariate, Sequential, Time series	Real	13,197	4	[52, 64, 124, 138]

Table 11 (continued)

Dataset name	Repository	Dataset characteristics	Attribute characteristics	Number of instances	Number of attributes	Ref#
3D Road Network (North Jutland, Denmark) data set	UCI	Sequential, Text	Real	434,874	4	[52, 114]
Coronary illness	UCI	NA	NA	175	29	[120]
Water treatment plant	UCI	Multivariate	Integer, Real	527	38	[52, 64, 120, 124, 138]
Housing	UCI	Multivariate	Numeric, Categorical	506	14	[52]
Cloud	UCI	Multivariate	Real	1,024	10	[52]
Twitter data set for Arabic sentiment analysis	UCI	Text	NA	2,000	2	[52, 64, 124, 138]
Localization Data for Person Activity Data Set	UCI	Univariate, Sequential, Time-Series	Real	164,860	8	[120]
Energy dataset	NA	NA	NA	252,447,882	10	[108]
Traffic dataset	NA	NA	NA	12,828	5	[108]
Detection of IoT botnet attacks N-BaIoT	UCI	Multivariate, Sequential	Real	7,062,606	115	[136, 145]
Cover type data set	UCI	Multivariate	Categorical, Integer	581,012	54	[159]
Iris data set	UCI	Multivariate	Real	150	4	[119]
Wine data set	UCI	Multivariate	Integer, Real	178	13	[119]
Yeast data set	UCI	Multivariate	Real	1,484	8	[119]
DLePM	Real industrial data sets	NA	NA	500,000	52	[119, 145]
sIoT	Real industrial data sets	NA	NA	6,000,000	255	[119, 145]
Poker Hand data set	UCI	Multivariate	Categorical, Integer	1,025,010	11	[145]
SUSY Data Set	UCI	NA	Real	5,000,000	18	[145]
Artificial data set 1	Scikit-learn	NA	NA	10,000	100	[135]
Artificial data set 2	Scikit-learn	NA	NA	12,000	80	[135]
Artificial data set 3	Scikit-learn	NA	NA	8000	60	[135]
Artificial data set 4	Scikit-learn	NA	NA	7500	100	[135]
Teaching assistant evaluation data set	UCI	Multivariate	Categorical, Integer	151	5	[123]
Contraceptive method choice data set	UCI	Multivariate	Categorical, Integer	1473	9	[123]

Table 12 Datasets details

Dataset Name	Website Address	Size	Ref#
Daily generation capacity, PJM	http://dataminer2.pjm.com/feed/day gen capacity	NA	[85]
Open energy information	https://openei.org/datasets/dataset/commercial-and-residential-hourlyload-profiles-for-all-tmy3-locations-in-the-united-states	NA	[85]
Accu weather incorporation	http://www.accuweather.com/en/us/united-states-weather	NA	[85]
Trans-African Hydro-Meteorological Observatory (TAHMO)	https://tahmo.org/	NA	[105]
University of KwaZulu-Natal-UKZN weather station	http://agromet.ukzn.ac.za:5355/index.html	NA	[105]
Kolumbus real time open data	https://opencom.no/	NA	[105]
Fire department calls for service dataset	https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3	NA	[162]
San Francisco open data portal	https://datasf.org/opendata	NA	[162]
Dataset Water meters	http://data.surrey.ca/dataset/water-meters	– 5 MB [51, 91] – 4 MB [63]	[51, 55, 63, 74, 91]
Traffic dataset	http://iot.ee.surrey.ac.uk:8080/datasets.html#traffic	3.04 GB	[55, 63]
Parking lots dataset	http://iot.ee.surrey.ac.uk:8080/datasets.html#parking	– 294 KB [51] – 0.20 MB [63]	[51, 55, 57, 63, 74, 91]
Pollution dataset	http://iot.ee.surrey.ac.uk:8080/datasets.html#pollution	– 32 GB + 570 MB [51, 53] – 77.25 MB [63]	[51, 53, 55, 63, 74]
Smart home dataset	Harvard Education website (https://dataverse.harvard.edu)	NA	[56]
Center for Applied Internet Data Analysis (CAIDA)	https://www.caida.org/catalog/datasets	NA	[111]
Floods	http://floodobservatory.colorado.edu/Archives/index.html	16 MB	[51]
Madrid highway vehicular traffic	http://trullols.site.ac.upc.edu/downloads	450 MB	[51, 74, 91]
Vehicular mobility traces	https://sourceforge.net/projects/sumo/	4.03 GB	[51, 74, 91]
Aarhus city traffic	http: //www.odaa.dk/dataset/realtids-trafikdata	33 GB	[51, 66, 74, 159]
Weather	http: //cdr.eionet.europa.eu/ro/eu/eiodata	3 MB	[51, 74]
Floods	http://floodobservatory.colorado.edu/Archives/index.html	16 MB	[51]
Flood disaster	https://www.indiawaterportal.org/	NA	[75]
Intel Berkeley research laboratory	http://db.csail.mit.edu/labdata/labdata.html	NA	[65]
Fire	NA	500 MB	[53]
Traffic	https://github.com/volkhin/RoadTraf_cSimulator	400 MB	[53]
Twitter	https://archive.org/details/twitterstream	41 GB	[53, 91]
STL-10 (image recognition dataset)	https://cs.stanford.edu/ ~ acoates/stl10/	1300 images (500 train data 800 test data)	[163]
NUS-WIDE-14 (image dataset)	https://lms.comp.nus.edu.sg/research/NUS-WIDE.htm	269,648 images	[161]
CUAVE (audio and image information)	[164]	7000 utterances	[163]

Table 12 (continued)

Dataset Name	Website Address	Size	Ref#
SNAE2 (subject from YouTube)	[165]	1800 pieces	[161, 163]
Real-life paths in a smart city	https://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html	NA	[166]
Shanghai futures exchange and Dalian futures exchange	http://epic.hust.edu.cn/download/Trade%20Dataset	NA	[9]
Dataset loading utilities	https://scikit-learn.org/stable/datasets/	NA	[114]
Ipv4 routed/24 topology dataset	http://www.caida.org/data/active/ipv4routed24topologydataset.xml	NA	[117]

processing, management, and service. The data collection layer includes data sensing, acquisition, buffering, and filtration. The big data are divided into small pieces in the processing layer, processed in parallel using HDFS and MapReduce, and stored. The management layer uses medical expert systems for processing the results and recommending corresponding actions.

Chui et al. [126] proposed a 6-layer architecture for patient behavior monitoring based on big data and IoT. Message queue, Apache Hadoop, behavior analytics, Mongo database, distributed stream processing, and exposer are the layers of this architecture. This architecture uses Hadoop for processing (descriptive, diagnostic, predictive, and prescriptive analytics), MongoDB for storing, Spark/Flink/Storm for stream processing, and Apache Kafka for breaking up the data stream into several partitions. Also, the authors have discussed the challenges of trust, security, privacy, and interoperability in the healthcare research field.

Ullah et al. [140] proposed a lightweight Semantic Interoperability Model for Big-Data in IoT (SIMB-IoT). The SIMB-IoT model has two main components: user interface and semantic interoperability. The semantic interoperability component is divided into three subcomponents: semantic interoperability, cloud services, and big data analytics. IoT data is collected and directed into an intelligent health cloud for online storage and processing. After processing, it sends suitable medicines to the patient's IoT devices. This article used the SPARQL query to find hidden patterns.

Elhoseny et al. [173] presented a Parallel Particle Swarm Optimization (PPSO) algorithm for IoT big data analysis in cloud computing healthcare applications. This article aims are: optimize virtual machine selection and storage by using GA, PSO, and PPSO algorithms; real-time processing; and reducing the execution time. This architecture has four components: stakeholders' devices; tasks; cloud broker; and network administrator. The cloud broker sends and receives requests to the cloud. The network administrator finds the optimal selection of virtual machines in the cloud for task scheduling.

Manogaran et al. [141] proposed a secured cloud-fog-based architecture for storing and processing real-time data for health care applications. This architecture has two sub-architectures: meta fog-redirection and grouping and choosing architectures. The meta fog-redirection architecture has three phases: data collection, data transfer, and big data storage. The data collection phase collected data from sensors in fog computing. The data transfer phase used the 's3cmd utility' method for transferring data to Amazon S3.The big data storage phase used Apache Pig and Apache HBase for storage. The grouping and choosing architecture protects data and provides security services in fog and cloud environments. Also, this architecture used MapReduce to predict.

García-Magariño et al. [156] is an agent-based simulation framework for IoT BDA in smart beds. This framework has two layers: the primary mechanism for simulating sleepers' postures and the information's analyzer. The first layer provides the simulation of the poses of sleeper mechanisms. The second layer analysis collected data from the first layer. The agent types in this framework are sleeper agent, weight sensor agent, bed agent, observer agent, analyzer agent, stochastic sleeper agent, bed sleeper agent, restless sleeper agent, and healthy sleeper agent. This framework helps researchers to test different sleeper posture recognition algorithms, discusses other sleeper behaviors, and performs online or offline detection mechanisms.

Yacchirema et al. [59] proposed a 3-layer architecture for sleep monitoring based on IoT and big data at the network's edge. The layers of this architecture are the IoT layer, the fog layer, and the cloud layer. The IoT layer collected and aggregated the big data and directed them to the fog layer. The fog layer is responsible for connectivity and interoperability between heterogeneous devices, preprocessing the collected data, and sending notifications to react in real-time. The big data is stored, processed, and analyzed in the cloud layer for intelligent decision-making. This layer has three modules: data management, big data analyzer, and web application. This architecture used HDFS for data storage and Spark for offline and real-time processing.

BigReduce [137] is a cloud-based IoT framework for big data reduction for health monitoring in smart cities that focuses on reducing energy costs. This framework has two schemes: real-time big data reduction and intelligent big data decision-making. The big data reduction is made in two phases: at the time of acquisition and before transmission using an event-insensitive frequency content process.

Ma et al. [33] proposed a 3-layer architecture for the IoT big health system based on cloud-to-end fusion. The layers of this architecture are the big health perception layer, transport layer, and big health cloud service layer. In the big health perception layer, data are collected and preprocessed. The transport layer sends data to sensor nodes and receives data from the perception layer using network technologies. The big health cloud service layer has two sub-layers: the cloud service support and the cloud service application. The cloud service support sub-layer is responsible for compressing, storing, processing, and analyzing the real-time data. The cloud service application sub-layer is the interface between users and health networking. This sub-layer controls the sensor nodes and visualizes the big data.

Rathore et al. [61] proposed the 5-layer architecture for big data IoT analytics-based real-time medical emergency response systems. The data collection layer is responsible for data sensing, acquisition, buffering, filtration, and processing. This layer collected and aggregated data using a coordinator or relay node and transmitted them to a polarization mode dispersion. The communication layer provides device-to-device communication to various smart devices. The processing layer divides big data into small chunks. Each chunk is processed separately, aggregated, and stored. This article used MapReduce, HDFS, and Spark for data processing and analysis. The management layer is responsible for managing all types of outcomes using a medical expert system. The service layer is the interface between end-users and health networking. This architecture minimized the processing time and increased the throughput.

El-Hasnony et al. [84] proposed a hybrid real-time remote patient monitoring framework based on mist, fog, and cloud computing. This article provided the 5-layer architecture for near real-time data analysis. The layers are the perception layer, the mist layer, the fog layer, the cloud layer, and the service provider layer. The mist layer is responsible for data filtering, data fusion, anomaly detection, and data transmission to the fog layer. The fog layer has done local monitoring and analysis, data aggregation, local storage, data pre-analysis, and data transmission to the cloud layer. The cloud layer implemented several data analytics techniques for intelligent decision-making and storage. This article presented a case study comparing traditional data mining techniques, including REPtree, MLP, Naive Bayes (NB), and sequential minimal optimization algorithms. The results showed that the REPtree algorithm achieved better accuracy, and the NB achieved the least time.

Harb et al. [106] proposed the 4-layer architecture for real-time BDA for patient monitoring and decision-making in healthcare applications. The layers of this platform are real-time patient monitoring, real-time decision and data storage, patient classification, and disease diagnosis, and data retrieval and visualization. The first layer is responsible for data ingestion using Kafka and Sqoop tools. The second layer processes and stores data using Spark and Hadoop HDFS. This layer preprocesses data and finds the missing records using MissRec (a script for Spark). The third layer is responsible for classification data using stability-based K-means, an adapted version of K-means clustering, and disease diagnosis using a modified version of the association rule mining algorithm. The last layer retrieves and visualizes data to understand the patient's situation using Hive, SparkSQL, and Matplotlib.

Zhou et al. [62] proposed a data mining technology based on the IoT. The layers of the proposed functional architecture are the data acquisition layer, data transmission layer, data storage layer, and cloud service center layer. This article used the WIT120 system for data collection, the adaptive k-means clustering method based on the MapReduce framework for data preprocessing, HDFS for storing, and the GM (1,1) grey model for users' health status prediction.

Hong-Tan et al. [90] proposed a real-time Ambient Intelligence assisted Student Health Monitoring System (AmIHMS). The data required by time ambient intelligence environments are collected from the WSN and sent to the cloud for handling. Their work developed a framework for real-time effective alerting of student health information. The AmIHMS architecture has three layers. The IoT layer collects health data from medical devices and sensors and saves it on one mobile computer or smartphone. The cloud layer receives the data through internet platforms such as 4G, 5G, LTE, etc., and executes the mining algorithms to extract relevant data for processing. The student health monitoring layer performs four stages to provide information and warnings about student health status. These stages include data retrieval, preprocessing, normalization, and classification/health status recognition.

Li [30] designed the fog-based Smart and Real-time Healthcare Information Processing (SRHIP) system. SRHIP architecture has three layers. IoT body sensor network layer performs data collection (health, environment, and locality), aggregation, compression, and encryption. Fog processing and computation layer use Spark and Hadoop ecosystem for information extraction, data normalization, rule engine, data filtration, and data processing. This layer performs the classification using the NB classifier. The cloud computation layer performs in-depth data analysis, storage, and decision-making. SRHIP minimizes the delay, transmission cost, and data size. This article uses hierarchical symmetric key data encryption to increase confidentiality.

The Improved Bayesian Convolution Network (IBCN) was proposed for human activity recognition [87]. The system architecture includes Wi-Fi and clouds onboard applications. The combination of a variable autoencoder with a standard deep net classifier is used to improve the performance of IBCN. This article used the convolution layers to extract the features and Enhanced Deep Learning (EDL) for security issues. IBCN provided the ability to download data via traditional radio frequency or low-power back-distribution communication. According to the experimental analysis, the proposed method allows the network to be continuously improved as new training sets are added and distinguishes between data-dependency and model-dependency. This architecture has high accuracy, versatility, flexibility, and reliability.

Sengupta and Bhunia [88] implemented a 3-layer IoT-enabled e-health framework for secure real-time data management using Cloudlet. The IoT layer uses IoT Hub for communicating with IoT devices. The Cloudlet layer is an intermediate layer between the IoT and cloud layers. This layer performs in-depth healthcare data analytics and processes. The cloud layer performs various analytics applications and processes queries. This framework uses SQLite for data storage in IoT Hub and Cassandra for future storing of sensed data. The result demonstrated that this framework has high efficiency, low data transmission time, low communication energy, data-packet loss, and query response time.

IBDAM [133] is an Intelligent BDA Model for efficient cardiac disease prediction in the IoT using multi-level fuzzy rules and valuable feature selection. This article used the open-source UCI database. First, it performs preprocessing on the UCI database, and the next step uses multi-level fuzzy rule generation for feature selection. IBDAM uses an optimized Recurrent Neural Network (RNN) to train the features. Finally, the features are classified into labeled classes according to the risk of evaluation by a medical practitioner. The results of this article demonstrate that this architecture has high performance and is quick and accurate.

Ahmed et al. [158] proposed an IoT-based health monitoring framework for pandemic disease analysis, prediction, and detection, such as COVID-19, using BDA. In this framework, the COVID-19 data set is collected from different data sources. Four data analysis techniques are performed on these data, including descriptive, diagnostic, predictive, and prescriptive. The experts opine on the results, and then users receive the results of these analyses through the internet and cloud servers. This article uses a neural network-based model for diagnosing and predicting the pandemic. The results of this article indicated that the accuracy, precision, F-score, and recall of the proposed architecture are better than AdaBoost, k-Nearest Neighbors (KNN), logistic regression, NB, and linear Support Vector Machine (SVM).

Ahanger et al. [71] proposed an IoT-based healthcare architecture for real-time COVID-19 data monitoring and predicting based on fog and cloud computing. This architecture has four layers. The data collection layer collects data from sensors and uses protocols to guarantee information security. The information classification layer classifies the information into four classes: health data, meteorological data, location data, and environmental data. The COVID-19-mining and extraction layer is responsible for splitting information into two groups using a fuzzy C-means procedure in the fog layer. The COVID-19 prediction and decision modeling layer use temporal RNN for estimating the results of the COVID-19 measure and a self-organization map-based technique to increase the perceived viability of the model. This article, in contrast to the existing methods, has high classification efficiency, viability, precision, and reliability.

Oğur et al. [109] proposed a real-time data analytics architecture for smart healthcare in IoT. This architecture has two domains. The software-defined networking-based WSN and RFID technology are used in the vertical domain, and data analytics tools, including Kafka, Spark, MongoDB, and NodeJS, are used in the horizontal domain. The collected data from WSN using RFID transmit to the Kafka platform using TCP sockets. The Kafka sends data to three consumers: The Apache Spark analysis engine that analyzes data in real-time; the NodeJS web application that visualizes patient data; and the MongoDB database that stores data. This article uses logistic regression and Apache spark MLlib for data classification. The result demonstrated this architecture has high performance and accuracy and is appropriate for a time-saving experimental environment.

Table 13 shows the result of the analysis of the articles. This table shows each article's architecture or framework name, OS name, programming language, advantages, and disadvantages. As you can see, layered architecture is the most important, with 14 articles.

3.2.1.2 BDM architectural/framework for IoT-based smart cities

According to the United Nations forecasting, about 67% of the world population will live in urban areas by 2050, resulting in environmental pollution, ecosystem destruction, energy shortage, emission reduction, and resource limitation [36, 174, 175]. Smart cities are large-scale distributed systems that could be a solution to overcoming these problems and improving intelligent services for residents [112, 176]. Smart cities have many implemented sensing devices that generate large amounts of data. These data must be stored, processed, and analyzed to extract valuable information [177]. BDM plays a significant role in this context and facilitates better resource management and decision-making [176]. Many research focused on BDM mechanisms in IoT-based smart cities with different objectives, including improving monitoring and communication, real-time controlling, and increased quality attributes (such as reliability, throughput, energy conservation, accuracy, scalability, delay, bandwidth usage, etc.). Therefore, this subsection examines the articles (14 articles; 22.22%) that have discussed the architectures or frameworks of BDM in IoT-based smart cities.

Jindal et al. [85] propose a tensor-based big data processing technique for energy consumption in smart cities. This article aims to reduce the dimensionality of data and decrease the overall complexity. The proposed framework has two phases. The first phase is the 3-layer data gathering and processing architecture. The layers of this architecture are data acquisition, transmission, and processing. In the second phase, the collected data was represented in tensor form, and SVM was used to identify the loads to manage the demand response services in smart cities. The technique reduces data storage by 38%.

ESTemd [105] is a distributed stream processing middleware framework for real-time analysis using big data techniques on Apache Kafka. The layers of this framework are the data ingestion layer, the data broker layer (source), the stream data processing engine and services, the data broker layer (sink), and the event hub. The data broker layer is responsible for data processing and transformation, with the support of multiple transport protocols. The third layer does stream processing and consists of the predictive data analytics model and Kafka CEP operators. This framework helps with performance improvement through data integration and distributed applications' interoperability.

CPSO [115] is a self-adaptive preprocessing approach for big data stream classification. This approach handles four mechanisms: sub-window processing; feature extraction; feature selection; and optimization of the window size and feature picking. CPSO uses clustering-based PSO for data stream mining; the sliding window technique for data segmentation; statistical feature extraction for variable partitioning; correlation feature selection, and information gain for feature selection. The proposed approach improves its accuracy.

Rani and Chauhdary [72] proposed a novel approach for smart city applications based on BDA and a new protocol for mobile IoT. They presented the 5-layer architecture where the layers are: data source, technology, data management, application, and utility programs. The data source layer collects, compresses, and filters data. The technology layer is responsible for communication between sensor nodes, edge nodes, and base station. The management layer used MapReduce, SQL, and Hbase for analyzing, storing, and processing. The utility program layer used WSN and IoT protocols to work with the other layers. Also, this article presented a new protocol that reduces energy consumption, increases throughput, and reduces the delay and transmission time.

SCDAP [107] is the 3-layer BDA architecture for smart cities. The first layer is the platform that includes hardware clusters, the operating system, communication protocols, and other required computing nodes. The second layer is security. The last layer is the data processing layer that supports online and batch data processing. This layer has ten components: data acquisition; data preprocessing; online analytics; real-time analytics; batch data repository; batch data analytics; model management; model

Table 13 Categories based on BDM architectural/framework for IoT-based healthcare

Architecture/ framework name	Architecture/framework model	OS name	Programming language	Advantages	Disadvantages
HICS [58]	Layered-based	UBUNTU	Java	-Real-time, offline, and parallel processing -Continuous monitoring -Provides feedback to users -Energy is conserved -Delay tolerant	-No comparison is made -The number of sensors is increased; the processing time is increased -The processing mechanisms are not specified -Does not support complex medical diagnoses -Overlooking data loading and ingestion into Hadoop
Chui et al. [126]	Layered-based	N/A	N/A	-High confidentiality	-Related work is not mentioned -The dataset is not specified -No comparison was made -No real-time processing
SIMB-IoT [140]	Component-based, cloud-based	N/A	N/A	-Lightweight model -Effective transfers of information from the patient to the physician	-The syntactic interoperability and security issues do not consider -Interoperability is only among heterogeneous IoT devices -Only one healthcare system is considered
Elhoseny et al. [173]	Component-based, cloud-based	N/A	MATLAB	-Minimizes delay and maximizes efficiency and resource utilization in the cloud -Parallel processing -Saves time and cost	-For applications that require overuse of processors to run functions on a cloud is not recommended -Does not work in different environments -Task scheduling process is not specified
Manogaran et al. [141]	Layered-based, cloud-based	N/A	N/A	-Providing security services -Low CPU usage and 72.82% accuracy -Reduces response time and error	-The obtained accuracy has not reached 100%
ABS-BedIoT [156]	Layered-based	N/A	N/A	-Average accuracy of 98% -Performs online or offline detection mechanisms	-No comparison was made -No test has been performed for a larger dataset
Yacchirema et al. [59]	Layered-based, Component-based, cloud/fog-based	Android & Contiki OS	Python	-Real-time and batch processing -High interoperability -93.3% effectiveness and the prediction error is 6.7%	-No comparison was made -Does not use other solutions applied to the healthcare domain -Evaluation with fewer patients
Wang et al. [137]	Cloud-based	TinyOS	C	-Process both the real-time and offline data -Reduces total energy and bandwidth cost -Provides an off-the-shelf and reliable solution	-No comparison was made -The dataset is not specified -Only focus on structural health events

Table 13 (continued)

Architecture/ framework name	Architecture/framework model	OS name	Programming language	Advantages	Disadvantages
Ma et al. [60]	Layered-based, Component-based, cloud-based	N/A	N/A	-Batch processing and streaming processing	-The dataset is not specified -No comparison was made -Fully evaluation has not been performed
Rathore et al. [61]	Layered-based	UBUNTU	N/A	-Supports real-time, offline, and batch processing	-No comparison was made A.A.A.A.A.C.-The use of Hadoop is not efficient for the small dataset (Less than 100 MB)
El-Hasnony et al. [84]	Layered-based, cloud/fog/mist-based	N/A	Java	-Accuracy of the REPtree algorithm is between 90.66% and 93.6%, according to the data's size -NB had the lowest time between 1 and 18 s -Near and real-time data analysis -High QoS and low end-to-end latency	-The dependence of the accuracy of the size of the data volume -Does not test on real-world problems
Harb et al. [106]	Layered-based	UBUNTU	Java & Python	-Batch and real-time processing -Fast computation process -Optimize the processing time speed up	-Clustering accuracy is decreased with the increase of the number of clusters -Had not to test in real-case scenarios -Does not cover mobile applications
Zhou et al. [62]	Layered-based, cloud-based	N/A	N/A	-Improves the accuracy -Reduces the computational load of the algorithm	-No comparison was made -The dataset is not specified -Fully evaluation has not been performed
Hong-Tan et al. [90]	Layered-based, cloud-based	N/A	MATLAB	-High reliability, accessibility, and accuracy -Real-time and historical data processing -Fast data reporting and data processing	-Fully evaluation has not been performed
Li [30]	Layered-based, cloud/fog-based	N/A	N/A	-Minimizes delay and data size and maximizes efficiency -Low transmission cost and communication-cost -Real-time and offline analysis -High security and throughput	-Only collected three data types: relevant to health, environment, and locality -Not considered the mobility of sensors
Zhou et al. [87]	Layered-based, cloud-based	N/A	N/A	-High accuracy, versatility, flexibility, and reliability -Reduces anomalous behaviors -Low power consumption	-Not detect the activity in real-time -Increasing the number of layers it increases the complexity of the architecture
Sengupta and Bhunia [88]	Layered-based, cloud-based	UBUNTU	N/A	-Real-time processing and near the IoT devices -Low-latency processing -Reduces the usage of communication resource -Energy is conserved	-Fully evaluation has not been performed -Not considered the failure issue in the cloudlet node -Not optimized the overall computation overhead and communication cost

Table 13 (continued)

Architecture/ framework name	Architecture/framework model	OS name	Programming language	Advantages	Disadvantages
Safa and Pandian [133]	Layered- based	N/A	Python	-Reduces the number of features -High accuracy and routing performance	-Related work is mentioned weakly -Fully evaluation has not been performed -How the preprocessing processes are not mentioned
Ahmed et al. [158]	Layered-based, cloud-based	N/A	N/A	-High accuracy, recall, and precision -Improves prediction accuracy	-Fully evaluation has not been performed -How the preprocessing processes are not mentioned
Ahanger et al. [71]	Layered-based, cloud/fog-based	Windows 10	MATLAB and Python	-Real-time processing -High classification efficiency, viability, precision, and reliability -Improves prediction accuracy -Low latency time and response delay	-How the preprocessing processes are not mentioned
Oğur et al. [109]	Layered-based	UBUNTU	N/A	-Real-time processing -High accuracy and performance -Low scheduling delay	-Does not use cloud services -Fully evaluation has not been performed

aggregation; smart application; and user interface. This architecture used Hadoop and Spark for data analysis. Also, this article presented a taxonomy of literature reviews based on six characteristics: focus, goal, organization, perspective, audience, and coverage.

Chilipirea et al. [80] proposed a data flow-based architecture for big data processing in smart cities. The architecture has seven steps: data sources, data normalization; data brokering; data storage; data analysis; data visualization; and decision support systems. This article used Extract, Transform, and Load (ETL) and Electronic Batchload Service (EBS) for normalizing the real-time and batch data. The data brokering step created the links between the collected data and the relevant context. This architecture used Hadoop for batch data processing and Storm for real-time data processing.

Gohar et al. [92] proposed a four-layer architecture for analyzing and storing data on the Internet of Small Things (IoST). The layers of this architecture are the small things layer, the infrastructure layer, the platform layer, and the application layer. The first layer collected data by using the LoRa gateway from LoRa devices. The infrastructure layer provides connectivity to devices by using the Internet. The platform layer is responsible for data preprocessing. For processing, this layer employs Max–Min normalization, the Kalman filter, the Round-Robin load balancing technique, the Least Slack Time algorithm (LST), the divide-and-conquer approach for aggregation, and NoSQL databases for storage. In the last layer, data is visualized for decision-making. This article implemented the architecture by using Hadoop, Spark, and GraphX. In this article, throughput has increased with the rise in data size.

Farmanbar and Rong [113] proposed an interactive cloud-based dashboard for online data visualization and a data analytics toolkit for smart city applications. The proposed architecture has three layers: the data layer, application and analysis layer, and presentation layer. The data layer is the core of the architecture and contains data acquisition units, data ingestion, data storage, and data access. This architecture used Logstash for data ingesting, Elasticsearch for storing, and Kibana for accessing and real-time monitoring. This platform has been tested on five datasets, including transportation data, electricity consumption, cargo e-bikes, parking, vacancies, and energy. The results showed this architecture is robust, scalable, and improves communication between users and urban service providers.

He et al. [116] proposed a big data architecture to achieve high Quality of Experience (QoE) performance in smart cities. This architecture has three plans: the data storage plane, the data processing plane, and the data application plane. This article used MongoDB and HDFS for data storing and Spark and the deep-learning-based greedy algorithm for data processing. The simulation result indicated that the proposed architecture's accuracy, precision, and recall are better than SVM and KNN.

Khan et al. [128] proposed an SDN-based 3-tier architecture that includes data collection, data processing and management, and an application layer for real-time big data processing in smart cities with two intermediate levels that work on SDN principles. This architecture uses Spark and GraphX with Hadoop for offline and real-time data analysis and processing. Also, this article proposed an adaptive job scheduling mechanism for load balancing and achieving high performance. The results showed that when clusters and processing time increase, the proposed system's performance also increases.

IoTDeM [73] is the IoT big data-oriented multiple edge-cloud architectures for MapReduce performance prediction with varying cluster scales. This architecture consists of three parts: multiple edge cloud redirectors, an edge cloud-based big data platform, and a centralized cloud-based big data platform. This architecture used historical job execution records and Locally Weighted Linear Regression (LWLR) techniques for predicting jobs' executing times and Ceph for storing them. Because of Ceph, there was no need to transfer data to the newly added slave node. This article validated the accuracy of the proposed model by using the TESTDFSIO and Sort benchmark applications in a general implementation scenario based on Hadoop2 and Ceph and achieved an average relative error of less than 10%.

Ahab [112] is a generic, scalable, fault-tolerant, and cloud-based framework for online and offline big data processing. This framework has four components: the user API, repositories, messaging infrastructure, and stream processing. The API directs the published data streams from different sources. Ahab uses the component, stream, policy, and action repositories for storing data streams, management policies, and actions. Ahab uses distributed messaging for handling data streams, minimizing unnecessary network traffic. Also, it allows the components to choose an appropriate communication point freely. The Ahab architecture has two layers: the streaming and service layers. The streaming layer is implemented as a lambda architecture. This layer has three sub-layers for data stream processing: the batch layer, the speed layer, and the serving layer. The HDFS and Apache Spark are used for data storing and stream processing. The service layer is responsible for analyzing, managing, and adapting components.

Mobi-Het [81] is a mobility-aware optimal resource allocation architecture for remote big data task execution in mobile cloud computing. This article uses the SMOOTH random mobility model to propound the free movement of mobile devices and estimate their speed and direction. Mobi-Het has three layers: mobile devices, cloudlets, and the master cloud. The mobile devices component has a decision-maker module that decides whether tasks should be executed remotely or locally. The master cloud component implements the resource allocation algorithm. This article has a low execution time, high execution reliability, and efficiency in timeliness.

Hossain et al. [132] proposed a knowledge-driven framework that automatically selects the suitable data mining and ML algorithms for a dynamic IoT smart city dataset. The system architecture has four units: data Knowledgeextraction, extactGoalKnowledge, extractAlgoKnowledge, and matchKnowledge. The framework's inputs are three key factors: datasets, goals, and data mining and ML algorithms. This article discussed both supervised and unsupervised data mining. The results show that this framework reduces computational time and complexity and increases performance and flexibility while dynamically choosing a high-accuracy solution.

Table 14 shows the result of the analysis of the articles. This table shows the architecture or framework name, OS name, programming language, advantages, and disadvantages of each article. As you can see, layered architecture is the most important, with 13 articles.

3.2.1.3 BDM architectural/framework for IoT-based smart home/ building

BDM mechanisms and IoT (architecture/ frameworks) have a crucial role in smart home/building, including processing data collected by the home sensors; analyzing, classifying, monitoring, and managing energy consumption and saving; intelligently identifying user behavior patterns and home activities; and increasing safety and comfort at home [76]. This subsection presents a review of the articles (8 articles; 12.70%) that have discussed the architectures or frameworks of BDM in the IoT-based smart home/ building.

Al-Ali et al. [68] proposed a smart home energy management architecture using IoT and BDA approaches. This architecture is divided into two sub-architectures: hardware architecture and software architecture. The hardware architecture includes sensors and actuators, high-end microcontrollers, and server blocks. The software architecture comprises the data acquisition module on the edge device, a middleware module, and a client application module. The first module monitors and collects data and transmits them to the middleware module. The second module uses several tools to provide different services, including facilitating communication between edge devices and middleware, data storage, data analysis, and sending results to the requester. The third module develops the front-end mobile user interface using a cross-platform integrated development environment. This article is

Table 14 Categories based on BDM architectural/framework for IoT-based smart cities

Architecture/ framework name	Architecture/ framework model	OS name	Programming language	Advantages	Disadvantages
Jindal et al. [85]	Layered-based, cloud-based	UBUNTU	N/A	-Accuracy of 99.2% -Reduces the data storage to 38%	-No comparison is made -Fully evaluation is not performed
ESTemd [105]	Layered-based, cloud-based	UBUNTU & MacOS Mojave	EPL	-Real-time, stream, and offline processing -Fast data integration -Low-latency processing -High throughput -Eliminates over-reliance on batch processing	-No comparison is made
CPSO [115]	Component-based	N/A	MATLAB & Java	-Offline and online analyzing -Self-adaptive -Has less costly than that the general PSO search -High ability to stochastic search -Accurate and efficient classification	-It is not better than the original and sliding window methods -High optimization time
Rani and Chauhdary [72]	Layered-based	N/A	MATLAB	-Energy is conserved -Low delay in big data gathering -Optimized in QoS -Increases network lifetime, scalability, and reliability	-Data is collected only by sensor nodes -All nodes are considered to be static -The dataset is not specified
SCDAP [107]	Layered-based	N/A	Python	-Real-time, online, and historical data analysis -Iterative and sequential data processing	-No evaluation is performed -Limited to Apache Hadoop suite as data storage and management layer -The dataset is not specified
Chilipirea et al. [80]	Layered-based	N/A	N/A	-Real-time and batch data processing -Resource limitations are flexible	-No comparison is made -It only focuses on the flow of data -Had not analyzed self-adaptive optimization methods -Does not focus on data preprocessing
Gohar et al. [92]	Layered-based	UBUNTU	N/A	-Online and offline processing -Throughput has increased with the rise in data size	-No comparison is made -The use of Hadoop is not efficient for the small dataset (Less than 100 MB) -The dataset is not specified
Farmanbar and Rong [113] (lighthouse project)	Layered-based, cloud-based	N/A	Python & JavaScript	-Offline and online processing and visualization -Self-service analytics -The cost of development or maintenance for the urban dashboard is meager -High scalability	-No comparison is made -Not used ML techniques -Evaluation metrics have not been examined -The big data mining process is not clearly explained

Table 14 (continued)

Architecture/ framework name	Architecture/ framework model	OS name	Programming language	Advantages	Disadvantages
He et al. [116]	Layered-based	N/A	N/A	-Real-time and offline analysis -Generates the highest QoE value	-The dataset is not specified
Khan et al. [128]	Layered-based	UBUNTU	N/A	-Real-time and offline processing -Efficiently balance the load on the Hadoop ecosystem -High accuracy and low execution time	-No comparison is made -Depends on the data size and the collected information/feature from the data sets -The dataset is not specified -Low security and feasibility
IoTDeM [73]	Layered-based, cloud-based	CentOS & UBUNTU	N/A	-Average relative error is less than 10% -Effectively predicts the total execution time of MapReduce applications -Reduces the bandwidth and delay -Avoiding the occurrence of the bottleneck -Scalable and flexible	-No comparison is made
Ahab [112]	Component-based, cloud-based	UBUNTU	Java & MONINA	-Scalable, fault-tolerant, flexible, evolvable, and generic -Performs online and offline analyses -Provides an extensible API -Minimizes unnecessary network traffic	-Not considering other challenges in the smart cities -Not applying ML techniques -The datasets are not specified
Mobi-Het [81]	Layered-based, cloud-based	Android	N/A	-Balances workload distribution among the cloudlets -Minimized total execution time, scheduling delay, and deviation of load distribution among the cloudlets -High reliability and stability -Heterogeneous-aware architecture	-No parallel execution -The dataset is not specified -High cost
Hossain et al. [132]	Layered-based	N/A	Python	-High flexibility and accuracy -Selects the best-suited data mining algorithms -Dynamic IoT data mining -Reduces the complexity and processing time -Considers both supervised and unsupervised algorithms for data mining	-Not considering scalability

evaluated using a prototype. The results showed the proposed architecture has high scalability, security, privacy, throughput, and speed.

Silva et al. [55] proposed a real-time BDA embedded architecture for the smart city with the RESTful web of things. This article integrated the web and smart control systems using a smart gateway system. The proposed architecture consists of four levels: data creation and collection; data processing and management; event and decision management; and application. The data processing and management level utilized HDFS for primary data storing, MapReduce for processing, Hbase to speed up the processing, and HIVE for data querying and managing. The event and decision management level classified two events as service and resource events based on the processed information. The application level remotely provides access to the smart city services and has three sub-layers: departmental layer, services layer, and sub-services layer. This article has high performance and throughput, low processing time, and minimizes energy consumption.

Khan et al. [57] proposed a scheduling algorithm, an IoT BDA architecture, and a real-time platform for managing sensors' energy consumption. This architecture has four steps: appliance discovery, sensor configuration and deployment, event management and scheduling, and information gathering and processing. Appliances are identified and classified in the first step based on user availability and usage time. The second step used Poisson distribution for sensor distribution in an IoT environment. In the third step, the appliance sleep-scheduling mechanism is presented for job scheduling. In the last step, the collected data from sensors were directed to Hadoop, Spark, and GraphX for processing and analysis. This step used HDFS for data storage. This article minimized total execution time and energy consumption.

HEMS-IoT [76] is a 7-layer architecture based on big data and ML for in-home energy management. The layers of this architecture are the presentation layer, IoT services layer, security layer, management layer, communication layer, data layer, and device layer. The management layer uses the J48 ML algorithm and the Weka API for energy consumption reduction and user behavior pattern extraction. This layer also classifies the data and houses based on energy consumption using the C4.5 algorithm. The IoT services layer provides different REST-based web services. The security layer guarantees data confidentiality. This layer has two components, namely authorization and authentication. This article uses RULEML and Apache Mahout to generate energy-saving recommendations.

Yassine et al. [56] proposed a platform for IoT smart homes based on fog and cloud computing. The components of the proposed platform are smart home components, IoT management and integration services, fog computing nodes, and cloud systems. The smart home component is divided into three tiers. The three tiers are: 1) the cyber-physical tier is responsible for interacting with the outside world through the second tier; 2) the connectivity tier is responsible for communicating with the smart home; and 3) the context-aware tier consists of user-defined rules and policies that create a privacy and security configuration. The IoT management and integration services component is in charge of providing interoperability, handling requests, authentication, and service registration. The fog computing nodes performed preprocessing, pattern mining, event detection, behavioral and predictive analytics, and visualization functions. The cloud system is responsible for storing and performing historical data analytics.

Luo et al. [131] proposed a 4-layer ML-based energy demand predictive model for smart building energy demands. Firstly, the sensitization layer collected data and transferred them to the storage layer. The storage layer performed data cleaning and storing. The model's smart core is in the analytics support layer, where Artificial Neural Network (ANN) and k-means clustering are used for identifying features in weather profile patterns. The service layer is an interface between the proposed model and the smart building management system. The proposed model improved accuracy and decreased mean absolute percentage error.

Bashir et al. [110] proposed an Integrated Big Data Management and Analytics (IBDMA) framework for smart buildings. The reference architecture and the metamodel are two phases of this framework. The reference architecture has eight layers: data monitoring, sourcing, ingestion, storage, analysis, visualization, decision-making, and action. People, processes, technology, information, and facility are the components of the metamodel phase. The core component of the metamodel is people (IoT policymakers, developers, and residents of intelligent buildings). The process component includes data monitoring, sourcing, ingesting, storage, decision-making, analytics, and action/control. The technology component consists of the tools and software packages to implement the IBDMA. Some of these tools are Apache Flume for data ingesting; HDFS for data storing; Apache Spark for data analysis; Microsoft Power BI for static data visualization; and Elasticsearch and Kibana for near-real-time data visualization. The information element manages disasters and controls various facilities based on results obtained by using the technology stack. The last element is the facility that improves the comfort, safety, and living conditions for the people of the building.

Table 15 shows the result of the analysis of the articles. This table shows each article's architecture or framework name, OS name, programming language, advantages, and disadvantages. As you can see, layered architecture is the most important, with five articles.

3.2.1.4 BDM architectural/framework for IoT-based intelligent transport

Safety, reliability, fault diagnosis, data transmission, and early warning in the intelligent transport system are critical for decision-making [178]. The intelligent transport system uses digital technologies, sensor networks, ML, and BDA mechanisms to overcome the challenges, including accident prevention, road safety, pollution reduction, automated driving, traffic control, intelligent navigation, and parking systems [179]. This subsection presents a review of the articles (2 articles; 3.17%) that have discussed the architectures or frameworks of BDM in IoT-based intelligent transport.

SMART TSS [129] is a BDA modular architecture for intelligent transportation systems. This architecture has four units: a big data acquisition and preprocessing unit, a big data processing unit, a big data analytics unit, and a data visualization unit. The big data processing unit stored the offline data in the cloud system for future analysis. The online data is sent to the extraction and filtration unit for load balancing on NoSQL databases. The big data analytics unit uses the map-reduce mechanism for analysis. This article uses Hadoop, Spark, and GraphX for big data processing and analysis. The throughput of the proposed system increases with increasing data size and has low accuracy and security.

Babar and Arif [89] proposed a real-time IoT big data analytics architecture for the smart transportation system. This architecture has three phases: big data organization and management, big data processing and analysis, and big data service management. The first phase performed data preprocessing, including big data detection, logging, integration, reduction, transformation, and cleaning. This phase used the divide-and-conquer technique for data aggregation, the Min–Max method for data transformation, and the Kalman filter technique for data cleaning. The second phase used Hadoop for big data processing, HDFS, Hive, and Hbase for data storage, and Spark for data stream analysis. This phase performed load balancing that caused increased throughput, minimized processor use, and reduced response time. The third phase is responsible for intelligent decision-making and event management.

Table 16 shows the result of the analysis of the articles. This table shows the architecture or framework name, OS name, programming language, advantages, and disadvantages of each article. As you can see, layered architecture is the most important, with two articles.

3.2.1.5 BDM architectural/framework for IoT-based traffic control and energy

Two reviewed articles discussed the architectures or frameworks of BDM in IoT-based traffic control and energy and used the ML for this purpose. ML4IoT [108] is a container-based ML framework for IoT data analytics and coordinating ML workflows. This framework aims to define and automate the execution of ML workflows. The proposed framework uses several types of ML algorithms. The ML4IoT framework has two layers: ML4IoT data management and ML4IoT core. The ML4IoT core layer trains and deploys ML models and consists of five components: a workflow designer, a workflow orchestrator, a workflow scheduler, container-based components, and a distributed data processing engine. ML4IoT data management is responsible for data ingesting and storing and has three sub-components: a messaging system, a distributed file system, and a NoSQL database. The results of this article reveal that this framework has high elasticity, scalability, robustness, and performance. Furthermore, Chhabra et al. [111] proposed a scalable and flexible cyber-forensics framework for IoT BDA analytics with high precision and sensitivity. This framework consisted of four modules: the data collector and information generator; feature analytics and extraction; designing ML models; and analyzing models on various efficiency matrices. This article used Google's programming model, MapReduce, as the core for traffic translation, extraction, and analysis of dynamic traffic features. Also, they presented a comparative study of globally accepted ML models for peer-to-peer malware analysis in mocked real-time.

Table 17 shows the result of the analysis of the articles. This table shows the architecture or framework name, OS name, programming language, advantages, and disadvantages for each article. As you can see, the component-based architecture is the most important, with two articles.

3.2.1.6 BDM architectural/framework for IoT-based urban planning

To improve the quality, plan, design, sustainability, living standards, dynamic organization, mobility of urban space and structure, and maintain the urban services, BDM is responsible for offline and online aggregation, managing, processing, and analyzing the large amounts of big data in urbanization [180–182]. Rathore et al. [51] proposed the 4-layer IoT-based BDA architecture for smart city development and urban planning. The first layer generated, aggregated, registered, and filtrated data from various IoT sources. Using communication technologies, the second layer created communication between sensors and the relay node. The third layer used HDFS, Hbase, Hive, and SQL for storage; MapReduce for offline analysis; and Spark, VoltDB, and Storm for real-time analysis. The last layer is

Table 15 Categories based on BDM architectural/framework for IoT-based smart home/ building

Architecture/ framework name	Architecture/ framework model	OS name	Programming language	Advantages	Disadvantages
Al-Ali et al. [68]	Component-based	N/A	JavaScript	-High scalability, security, and privacy -Real-time data analysis	-No comparison is made -The dataset is not specified
Silva et al. [55]	Layered-based	UBUNTU	C#	-Real-time data processing -Users can view and request information by using any web browser -Minimizes energy consumption based on user choices -High reliability data processing speed	-Not applying ML techniques -Fully evaluation has not been performed
Moreno et al. [69]	Layered-based	N/A	EPL	-Energy conservation -Faster convergence of the filter -Overcomes the challenge of velocity	-Not mentioned volatility
Khan et al. [57]	Layered-based	N/A	N/A	-Real-time and offline processing -Energy conservation -Minimizes total execution time	-No comparison is made -The dataset is not specified -Implement real sensors and hardware is not considered
HEMS-IoT [76]	Layered-based	Android	Java	-High scalability -Reduces energy consumption -Real-time data processing -Provides facilitates communication between devices and end users -High confidentiality	-No comparison is made -The dataset is not specified -Compatible with some types of home sensors -Only uses the J48 ml algorithm -Energy-saving recommendations are not customized by the system
Yassine et al. [56]	Layered-based, cloud/fog-based	N/A	Python	-Near real-time, online, and offline data processing -Handles large volumes of unstructured data -Minimizing communication overhead and latency -Ensures the privacy of the smart home -Facilitating various energy management programs	-No comparison is made -Fully evaluation has not been performed -The optimal distribution and configuration of fog nodes do not determine -Has not tested various datasets
Luo et al. [131] (Building heating and cooling)	Layered-based	N/A	N/A	-Improves prediction accuracy -The MAPE of energy demands prediction is 3% and 8% in training and testing cases -Low computation time	-No comparison is made -How to store and related tools are not specified -Only historical data has been considered
Bashir et al. [110]	Layered-based, cloud-based	N/A	Python	-Real-time and batch data analysis -It is based on the meta model	-Low applicability -There is no feedback mechanism

Table 16 Categories based on BDM architectural/framework for IoT-based intelligent transport

Architecture/ framework name	Architecture/framework model	OS type	Programming language	Advantages	Disadvantages
SMART TSS [129]	Layered-based, cloud-based	UBUNTU	N/A	-Offline and online data processing -Throughput has increased with the rise in data size -Produces good real-time results -The proof-of-concept implementation is encouraging	-The use of Hadoop is not efficient for the small dataset (Less than 100 MB) -The database used is not specified precisely -Low accuracy and security
Babar and Arif [89]	Layered-based	UBUNTU	N/A	-Real-time (stream) and offline data analysis -High throughput -Throughput has increased with the rise in data size	-No comparison is made -The dataset is not specified -The use of Hadoop is not efficient for the small dataset

Table 17 Categories based on BDM architectural/framework for IoT-based traffic control and energy

Architecture/framework name	Architecture/framework model	OS type	Programming language	Advantages	Disadvantages
ML4IoT [108]	Layered-based, component-based	N/A	Java	-Handles high volume of IoT data -Supports batch and online ML workflow	-Have not automated selection and tuning of ML models -Does not use cloud services
Chhabra, et al. [111]	Component-based	UBUNTU	Java & Python	-Sensitivity of 99% -High availability	-Only have used the gain ranking algorithm -Data security issue is not considered -Does not use cloud services

responsible for showing the study results for intelligent and fast decision-making. The results show that the architecture provides efficient outcomes even on IoT big data sets. Throughput has increased with the rise in data size, and the processing time has decreased.

Silva et al. [63] proposed a reliable 3-layer BDA-embedded architecture for urban planning. The layers of this architecture are data aggregation, data management, and service management. The purpose of this article is to increase throughput and minimize processing time. The real-time data management layer is the main layer and performs data filtration, analysis, processing, and storing. This layer used data filtration and min–max normalization techniques to improve energy data. This architecture used MapReduce for offline data processing, Spark for online data processing, and Hbase for storing.

Table 18 shows the result of the analysis of the articles. This table shows the architecture or framework name, OS name, programming language, advantages, and disadvantages for each article. As you can see, layered architecture is the most important, with two articles.

3.2.1.7 BDM architectural/framework for other IoT-based applications This subsection presents a review of the articles (14 articles) that have discussed the architectures or frameworks of BDM in other IoT-based applications. These IoT applications are smart IoT systems (4 articles), smart flood (1 article), smart farms (2 articles), disaster management (1 article), laundry (1 article), smart pipeline (1 article), network traffic (1 article), digital manufacturing (1 article), smart factory (2 articles).

Al-Osta et al. [121] proposed an event-driven and semantic rules-based approach for IoT data processing. The main levels of this system are sensor, edge, and cloud levels. This article has two purposes: reducing the required resources and the volume of data before transfer to the cloud for storage. The collected data is first aggregated, filtered, and classified at the gateway level. This causes a saving in bandwidth and minimizes the network traffic. This approach used semantic rules for data filtering. It also employed a complex event processing module to analyze input events and detect processing priority.

Wang et al. [148] proposed a 3-layer edge-based architecture and a dynamic switching algorithm for IoT big data analytics. The layers of this architecture are the cloud layer, edge layer, and IoT layer. The edge layer performed some functions, including identifying IoT applications, classifying them, and sending classification results to the cloud layer. The LibSVM method is used for IoT application identification and classification based on system status and requirements. Also, this article presented a new algorithm, namely the dynamic switching algorithm, for task offloading from cloud to edge based on the delay and network conditions. This algorithm performed task offloading based on classification results. The results showed the proposed architecture reduced delay, processing time, and energy consumption.

IODML-BDA [124] is a model for Intelligent Outlier Detection in Apache Spark using ML-powered BDA for mobile edge computing. This model performs four steps: data preprocessing, outlier detection, feature selection, and classification. This article employs an Adaptive Synthetic Sampling (ADASYN)-based technique for outlier detection, the Oppositional Swallow Swarm Optimization (OSSO) for feature selection, and a Long Short-Term Memory (LSTM) model for classification. This model has high performance and accuracy in BDA.

Kumar et al. [3] presented a novel 4-layer architecture for IoT big data management in cloud computing networks and a collaborative filtering recommender system. The information layer collects data and transmits them to the second layer. The transport layer uses GPRS/CDMA, wireless RFID, or Ethernet channels for communication and data uploading in the data mining layer. The data mining layer utilizes the ML method for data analysis. The application layer is responsible for data visualization based on extracted information from the data mining layer. The article also proposed a collaborative filtering algorithm to improve the prediction accuracy based on the time-weighted decay function and asymmetrical influence degree. The result of this article demonstrated that this architecture has high accuracy.

Sood et al. [75] proposed a 4-layer flood forecasting and monitoring architecture based on IoT, High-Performance Computing (HPC), and big data convergence. The IoT layer is responsible for IoT device installation and data collection. The fog computing layer reduces the latency of application execution when predicting the real-time flood. The data analysis layer received, stored, and analyzed the collected data. This layer used Singular Value Decomposition (SVD) for data reduction and a K-mean clustering algorithm to estimate the flood situation and rating. Also, Holt-Winter's forecasting method is utilized to forecast the flood. The last layer is the presentation layer, which generates information for decision-making. The results showed the proposed architecture reduced latency, complexity, completion time, and energy consumption.

Muangprathub et al. [79] proposed a WSN system for agriculture data analysis based on the IoT for watering crops. This system consists of three components. The hardware component collected data and sent them to the web application for real-time analysis. This component is responsible for data preprocessing, data reduction by the equal-width histograms technique, data modeling/discovery by association rules mining technique, and solution analysis. The web application manages real-time information. The mobile application component controlled crop watering remotely. The architecture of this system has three layers: the environmental data acquisition layer, the data, and communication layer, and the application layer. This system can help to reduce costs and increase agricultural productivity.

Al-Qurabat et al. [65] proposed a two-level system for data traffic management in smart agriculture based on compression and Minimum Description Length (MDL) techniques. The first level is the sensor node level. This level monitors the features of the environment using a lightweight lossless compression algorithm based on Differential Encoding (DE) and Huffman techniques. The second level is the edge gateway level. This level is responsible for processing, analyzing, filtering, storing, and sending the data to the cloud, and minimizes the first level dataset using MDL and hierarchical clustering. The results demonstrated the suggested method has a high compression ratio and accuracy and decreases data and energy consumption.

Shah et al. [53] proposed the 5-layer architecture for IoT BDA in a disaster-resilient smart city. The purpose of this architecture is to store, mine, and process big data from IoT devices. This architecture's layers include data resource, transmission, aggregation, analytics and management, and application and support services. This architecture used Apache Flume and Apache Sqoop for unstructured and structured data collection; Hadoop and Spark for real-time and offline data analysis; and HDFS for data storage. The proposed implementation model comprises data harvesting, data aggregation, data preprocessing, and a big data analytics and service platform. This article used a variety of datasets for validation and evaluation based on processing time and throughput.

Liu et al. [14] proposed a cloud laundry business model based on the IoT and BDA. This model used big data analytics, intelligent logistics management, and ML techniques for big data analytics. This model minimized human interference and increased system efficiency.

Tang et al. [7] proposed the 4-layer distributed fog computing-based architecture for big data analysis in smart cities. The layers of this architecture are the data center on the cloud layer, intermediate computing nodes layer, edge devices layer, and sensing networks on the critical infrastructure layer. This architecture reduces the communication bandwidth and data size. First, data was collected from the fiber sensor network and transmitted to the edge computing nodes layer. This layer performed two tasks: identifying potential threat patterns and feature extraction using supervised and non-supervised ML algorithms. The intermediate computing nodes layer used the hidden Markov model for big data analysis and hazardous event detection. The results showed the proposed architecture reduced the

Table 18 Categories based on BDM architectural/framework for IoT-based urban planning

Architecture/framework name	Architecture/framework model	OS type	Programming language	Advantages	Disadvantages
Rathore, et al. [51]	Layered-based	UBUNTU	N/A	-Offline and online data processing -More scalable and efficient than the existing systems -Throughput has increased with the rise in data size	-No comparison is made -Avoiding the scalability option for analyzing -Not mentioned in the relevant article
Silva, et al. [63]	Layered-based	UBUNTU	N/A	-Offline and online data processing -Applicability and reliability in the real world -Reduce processing time for MapReduce and Spark	-Not used ML techniques

service response time and the number of service requests submitted to the cloud.

Kotenko et al. [136] introduced a framework for security monitoring mobile IoT based on big data processing and ML. This framework consists of three layers: 1) extraction and decomposition of a data set using the heuristic approach; 2) compression of feature vectors using Principal Component Analysis (PCA); and 3) learning and classification using the SVM k-nearest neighbor's method, Gaussian NB, artificial neural network, and decision tree. This framework has high performance and accuracy in the detection of attacks.

Bi et al. [157] proposed a new enterprise architecture that integrates IoT and BDA for managing the complexity and stability of the digital manufacturing system. This article used Shannon entropy to measure the complexity of a system based on the number of events and the probabilities of event occurrences. This architecture performs three processes: data acquisition, management, and utilization. The result of this article demonstrated that this architecture decreases the system complexity and increases flexibility, resilience, responsiveness, agility, and adaptability.

Yu et al. [118] presented a BDA and IoT-based framework for health state monitoring in a smart factory. This framework consists of four phases. The data ingestion phase is responsible for extracting different data types, managing data collection, data security, data transformation using a secure file transfer protocol, and data storage issues. The big data management phase uses optimized HDFS for data storage on the cloud nodes and processing using Apache Spark. The data preparation phase performs sensor selection and noise detection processing to produce high-quality data. This phase uses the high-variance feature removal method for feature selection and a novel method for noise detection. The predictive modeling phase has four stages: PCA model training, streaming anomaly detection, contribution analysis, and alarm sequence analysis.

Kahveci et al. [183] proposed a secure, interoperable, resilient, scalable, and real-time end-to-end BDA platform for IoT-based smart factories. The platform architecture has five layers and several components that perform data collection, data integration, data storing, data analytics, and data visualization. The layers of architecture are the control and sensing layer, the data collection layer, the data integration layer, the data storage and analytics layer, and the data presentation layer. All kinds of sensing and control activities are performed in the first layer. The data collection layer communicates with the first layer through a multi-node client/server architecture. The data integration layer uses the RESTful application program interface to transfer data collected to the data storage layer. The data storage layer uses InfluxDB for industrial metrics and events. Using this architecture, production line performance is improved, bottlenecks are identified, product quality is improved, and production costs are reduced.

Table 19 shows the result of the analysis of the articles. This table shows the architecture or framework name, OS name, programming language, advantages, and disadvantages for each article. As you can see, layered architecture is the most important, with 14 articles.

3.2.2 BDM architectural/framework for IoT paradigms

Another category presented in this article is BDM architectures and frameworks in two important IoT paradigms, i.e., Social Internet of Things (SIoT) and Multiple Internet of Things (MIoT). SIoT is the integration of the IoT with social networking that leads to improved scalability in information and service discovery, trustworthy relationships, security, performance, and high network navigability [91, 184]. The SIoT establishes relationships and interactions between human-to-human, human-to-object, and object-to-object social networks in which humans are considered intellectual and relational objects [185, 186]. The types of relationships.

between smart, complex, and social objects in SIoT are parental object relationships, co-location object relationships, co-work object relationships, ownership object relationships, social object relationships, stranger object relationships, guest object relationships, sibling object relationships, and service object relationships [187, 188]. A MIoT is a collection of connected things that are different kinds of relationships and objects.

In contrast to SIoT, the number of relationships in MIoT is not predefined. Therefore, SIoT is a specific case of MIoT where the number of possible relationship types is limited [187]. The MIoT paradigm has advantages over the IoT and SIoT. IoT can be divided into multiple networks of interconnected smart objects through MIoT. The MIOT can handle situations where the same objects behave differently in different networks and allows objects from various networks to communicate without being directly connected [189]. Social objects in the SIoT and MIoT can perform tasks, including physical condition detection, data collection, information exchange, big data processing and analysis, and visualization for decision-making, predicting human behavior, and increasing efficiency and scalability. Due to the heterogeneous nature of communication and social networks, which generate high volume, multi-source, dynamic, and sparse data from SIoT and MIoT objects, the BDA is a vital issue in these paradigms. For BDA in SIoT and MIoT, a large amount of memory, power processing, and bandwidth are required to store, define, process, predict, and assist humans for a limited time

Table 19 Categories based on BDM architectural/framework for other IoT-based applications

Domain	Architecture/ framework name	Architecture/ framework model	OS type	Programming language	Advantages	Disadvantages
Smart IoT systems	Al-Osta, et al. [121]	Layered-based, Cloud-based	Raspbian	EPL & Java	-Reduces resources required for data processing -Reduces the amount of data to be transferred to the cloud -Real-time data processing	-No comparison is made -Not considered more compression parameters
	Wang, et al. [148]	Layered-based, Cloud-based	N/A	MATLAB	-Decreasing the average delay, processing time, and energy consumption -Low-complexity	-No comparison is made -The dataset is not specified
	IODML-BDA Mansour, et al. [124]	Layered-based	N/A	N/A	-High accuracy -Reduces energy and memory consumption	-Fully evaluation has not been performed -Not used ML techniques
	Kumar, et al. [3]	Layered-based, Cloud-based	N/A	MATLAB	-High accuracy -Decreases loss rate and recall rate	-Fully evaluation has not been performed -The ML algorithm used is not known
Smart flood	Sood, et al. [75]	Layered-based	Android & Windows	Java & MATLAB	-Reduces latency, complexity, and completion time -Reduces energy consumption -Efficiently for identifying appropriate locations to install IoT devices -Real-time prediction -Increases the efficiency of SVD	-No comparison is made -Criteria such as throughput, precision, sensitivity, specificity, etc., have not been evaluated
Smart farms	Muangprathub, et al. [79]	Layered-based, component–based,	N/A	N/A	-Monitoring and decision-making are improved -Real-time data processing -Increases productivity means -Carried out in a mixed crop environment -Both automatic and manual functional controls to the user	-No comparison is made -Fully evaluation has not been performed -Not used ML techniques -Generates huge volume of data
	Al-Qurabat, et al. [65]	Layered-based, Cloud-based	N/A	N/A	-Reduces the amount of data -Reduces energy consumption -High compression ratio and accuracy -Decreases algorithm complexity	-Does not reach the theoretical maximum -Not applying ML techniques
Disaster management	Shah, et al. [53]	Layered-based	UBUNTU	Java	-Real-time and offline analysis and processing -Work with different data sources	-Different data sources are not considered

Table 19 (continued)

Domain	Architecture/ framework name	Architecture/ framework model	OS type	Programming language	Advantages	Disadvantages
Laundry	Liu, et al. [14]	Layered-based, Cloud-based	N/A	Python	-Offline and real-time analysis, scheduling, and processing -Prevent human errors and increase the system efficiency -The total cost is lower than the traditional models -Improving the transparency and quality of laundry	-No comparison is made -Fully evaluation has not been performed -Does not use more computationally efficient Artificial Intelligence (AI) algorithms
Smart pipeline	Tang, et al. [7]	Layered-based, Cloud/fog-based	N/A	N/A	-Support multi-level data association -Suitable for large-scale and computing-intensive systems -Data and control flow -Reduces the transmission bandwidth, power consumption, and time to service	-No comparison is made -Not mentioned to distributed processing framework for fog -Not mentioned devices used as fog -The dataset is not specified
Network traffic	Kotenko, et al. [136]	Layered-based	N/A	N/A	-High performance and accuracy -Reduces the time costs for training	-No comparison is made -Real-time processing has not been reviewed -Not using special software such as Hadoop and Spark
Digital manufacturing	Bi, et al. [157]	Layered-based	N/A	N/A	-Reduces the system complexity -High flexibility, resilience, responsiveness, agility, and adaptability -Independent to products, sectors, or regions	-No comparison is made -Fully evaluation has not been performed
Smart factory	Yu, et al. [118]	Layered-based	N/A	N/A	-This framework is generic -Considers data quality issues -Real-time and batch analysis and processing	-No comparison is made -Fully evaluation has not been performed
	Kahveci, et al. [183]	Layered-based, component-based	N/A	N/A	-Scalable, secure, resilient, interoperable, and high performance -Reduces production cost -Batch and real-time data analysis	-No comparison is made -Fully evaluation has not been performed -Deploys only on on-premises clusters

[64, 91]. Different researchers have examined BDA in these paradigms in various ways.

Paul et al. [91] proposed a system called SmartBuddy that performs the BDA for SIoT-based smart city data to define real-time human dynamics. This architecture has three domains: the object domain, the SIoT server domain, and the application domain. The object domain collects the data and sends them to the SIoT server for balancing, storing, querying, processing, defining, and predicting human behavior. The application domain has four main components: security, cloud server, results in storage devices, and data server. This domain compilation is the result of the SIoT server domain. This article uses MapReduce programming for offline data analysis and Apache Spark for real-time analysis. SmartBuddy has high throughput and applicability.

HABC [52] is a Hadoop-based architecture for social IoT big data feature selection and analysis. This architecture has four layers: data collection, communication, feature selection and processing, and service. The data collection layer collected, registered, and filtered data. The communication layer provided end-to-end connectivity to various devices and used the Kalman filter to remove noise. The feature selection and processing layer used MapReduce for data analysis and HDFS, HBSE, and HIVE for manipulation and storing. The Artificial Bee Colony (ABC) is used for feature selection. The results indicate that the architecture increases throughput and accuracy and is more scalable.

Lakshmanaprabu et al. [64] proposed a hierarchical framework for feature extraction in SIoT big data using the MapReduce framework and a supervised classifier model. This framework has five steps: SIoT data collection, filtering, database reduction, feature selection, and classification. This article used the Gabor filter to reduce the noisy data, Hadoop MapReduce for database reduction, Elephant Herd Optimization (EHO) for feature selection, and a linear kernel SVM-based classifier for data classification. The result showed the proposed architecture has high maximum accuracy, specificity, sensitivity, and throughput.

Socio-cyber network [66] is the 4-layer architecture that integrates the social network with the technical network for analyzing human behavior using big data. This architecture uses the user's geolocation information to make friendships and graph theory to examine the trust index. The data generation layer is responsible for data collection, aggregating, registration, and filtration. The communication layer provides end-to-end connectivity to various devices. This layer creates a graph of data, and when new data are added to the system, this graph is updated. The data storage and processing layer perform the load balancing algorithm and graph processing. This layer uses MapReduce for data processing, the Spark GraphX tool for real-time analysis, and HDFS for data storage. This article uses the Knowledge Pyramid for knowledge extraction. The service layer shows the result to users.

Shaji et al. [120] presented a 5-phase approach for big data classification in SIoT. The phases of this approach are the data acquisition phase, data filtering phase, reduction phase, feature selection phase, and classification phase. This article uses an adaptive Savitzky–Golay filter for filtering and eliminating noisy data; the Hadoop MapReduce framework for data reduction; a modified relief technique for optimal feature selection; and a deep neural network-based marine predator algorithm for classification. This article has high accuracy, precision, specificity, sensitivity, throughput, and low energy consumption.

Floris et al. [67] proposed a 4-layer architecture based on SIoT to deploy a full-stack smart parking solution. The layers of this architecture are the hardware layer, virtualization layer, aggregation layer, and application layer. The hardware layer collected data and consisted of a vehicle detection board, Bluetooth beacon, data transmission board, and concentrator. The SIoT paradigm is implements in the virtualization layer using device virtualization. ML algorithms are implemented in the aggregation layer for data aggregation and data processing. The application layer includes the management platform that supports the control dashboard for smart parking management and the Android App for the citizens.

Cauteruccio et al. [166] presented a framework for anomaly detection and classification in MIoT scenarios. This framework investigated two problems: the anomaly effects analysis on the MIoT and the source of the anomaly detection. The anomalies in MIoT are divided into three categories: presence anomalies versus success anomalies, hard anomalies versus soft anomalies, and contact anomalies versus content anomalies.

Lo Giudice et al. [189] proposed a definition of a thing's profile and topic-guided virtual IoT. The profile of a thing has two components: a content-based component (past behavior) and a collaborative filtering component (principal characteristics of those things it has previously interacted with the most). This article uses a supervised and unsupervised approach to build topic-guided virtual IoTs in a MIoT scenario. Table 20 shows the result of the analysis of the articles. The architecture or framework name, the OS name, programming language, advantages, and disadvantages are shown for each article in this table. As you can see, layered architecture is the most important, with five articles.

3.3 Categories based on quality attributes

Systems have different attributes generally divided into qualitative or functional attributes and non-qualitative or

Table 20 Categories based on BDM architectural/framework for IoT paradigms

Architecture/framework/approach name	Architecture/framework model	OS name	Programming language	Advantages	Disadvantages
SmartBuddy [91]	layered-based	UBUNTU	NA	-High throughput and applicability	-No comparison is made -Fully evaluation has not been performed -Does not use cloud services -Not used ML techniques
HABC [52]	Layered-based	UBUNTU	Java	-Offline and online data processing -#selected feature decreased -High accuracy -Throughput has increased with the rise in data size	-Compared to some of the algorithms, it did not work well
Lakshmanaprabu, et al. [64]	Layered-based	Windows	Java	-Maximum accuracy, specificity, and sensitivity -Improves classification and efficiency -Increases Throughput vs. data size	-Real-time processing has not been reviewed -High process execution time
Socio-cyber network [66]	Layered-based	UBUNTU	NA	-High throughput, scalability, and efficiency -Low processing time	-No comparison was made -The use of Hadoop is not efficient for the small dataset (Less than 100 MB)
Shaji, et al. [120]	Component-based	Windows	Java	-High throughput, accuracy, precision, specificity, and sensitivity -Low energy consumption -Minimizing computational overheads	-Not considering scalability and security
Floris, et al. [67]	Layered-based	Android	Python	-High interoperability and performance	-No comparison was made -Fully evaluation has not been performed
Cauteruccio, et al. [166]	Component-based	UBUNTU	Python	-High throughput and applicability	-No comparison is made -Fully evaluation has not been performed -Does not use cloud services -Not used ML techniques
Lo Giudice, et al. [189]	Component-based	UBUNTU	Python	-With increasing the MIoT size, the number of clusters increases, but cluster size slightly increases -Low computation time -High efficiency	-No comparison is made -Fully evaluation has not been performed

non-functional attributes. This section considers the quality attributes of the selected articles. Quality attributes indicate the system's characteristics, operating conditions, and constraints. There are different software quality models, such as McCall [190], Bohem [191], ISO/IEC9126, and FURPS [192]. As far as we know, no systematic article has completely categorized articles based on qualitative characteristics. Therefore, this paper categorized the selected articles based on 18 qualitative attributes presented in Table 21. In this table, the first column shows the names of these 18 quality attributes. The reviewed articles used these quality attributes to show the characteristics, quality attribute analysis, and performance analysis of the proposed approaches, architectures, and frameworks and comparison with other works. Performance attributes have been analyzed in different articles based on different criteria. The reviewed articles utilized 12 quality attributes for performance attribute analysis. These quality attributes are load balancing, energy conservation, network lifetime, processing/execution time, response time, delay, CPU usage, memory usage, bandwidth usage, throughput, latency, and concurrency. In Table 21, ↓ indicates the reduction of that quality characteristic and ↑ indicates the increase of that quality characteristic. The second column in this table shows the articles that have used these features. The performance, efficiency, accuracy, and scalability attributes are the most critical quality attributes, with 79, 62, 58, and 47 articles, respectively. From another point of view, the reference model of standard software quality attributes, i.e., ISO 25010, has been used to classify articles based on quality attributes. Table 22 shows the articles' classification according to this standard. In the following, some quality attributes and their importance will be defined.

- Performance: Performance refers to the ability of BDM techniques in the IoT to provide results and services with high load balancing, energy conservation, throughput, concurrency, low processing/execution time, delay, CPU/memory/ bandwidth usage, and latency.
- Feasibility: Feasibility refers to the ability to perform successfully or study the current mode of operation, evaluate alternatives, and develop BDM techniques in the IoT.
- Scalability: Scalability refers to the ability of BDM techniques in the IoT to exploit increasing computing resources effectively to maintain service quality when the real data volumes increase. BDM techniques in IoT must be scalable in performance and data storage. Some methods and advanced systems are used to improve the scalability of big data analysis, like parallel implementation, HPC systems, and clouds [193].
- Accuracy: Accuracy refers to the ability to describe data and represent a real-world object or event correctly [194]. In the reviewed articles, various definitions of accuracy are provided, including clustering accuracy, classification accuracy, the accuracy of features selecting/extracting, and the accuracy of the prediction model. Each of these cases is evaluated in different ways.
- Efficiency: Efficiency refers to BDM techniques in IoT with minimum energy and response time and high throughput, accuracy, and performance.
- Reliability: Reliability refers to the ability of BDM techniques in the IoT to apply the specified functions under specified conditions and within the expected duration.
- Availability: The main goal of many researchers is the availability of information and their analysis from heterogeneous data sources. Availability is one of the components of service trust and is part of reliability.
- Interoperability: Interoperability refers to the ability to interconnect and communicate among smart objects, heterogeneous IoT devices, and different operating systems. Low-cost device interoperability is a vital issue in IoT [53, 54, 195].
- Flexibility: Flexibility refers to the capacity of BDM techniques in the IoT to be adapted for different environments and situations to face external changes [196].
- Robustness: Robustness refers to a stable BDM system in the IoT that can function despite erroneous, exceptional, or unexpected inputs and unexpected events.

3.4 Big data analytics types in IoT

There are different types of analytics. This study uses Gartner's classification,[2] which includes four types of analysis: descriptive analysis ("what happened?"), diagnostic analysis ("why did it happen?"), predictive analysis ("What could happen?"), and prescriptive analysis ("What should we do?"). In descriptive analytics, historical business data is analyzed to describe what happened in the past. Diagnostic analytics investigates and identifies the causes of trends and why they occurred. The goal of predictive analytics is to forecast the future using a variety of statistical and ML techniques. Prescriptive analytics proposes the best action to take to accomplish a business's objective using the data collected from descriptive and predictive analytics for decision-making based on future situations [197].

This paper investigates the applied methods for data analysis and categorizes them based on the type of analysis these methods provide. Organizations need statistics, AI,

[2] http://www.gartner.com/it-glossary/predictive-analytics/.

Table 21 Classification of articles based on quality attributes

Quality attributes			Articles
Performance	Load balancing	↑	[7, 57, 84, 112, 128, 129, 143, 144]
	Energy conservation	↑	[7, 30, 52, 55, 58, 61, 64, 67–69, 72, 75–77, 82, 86–88, 111, 120, 124, 125, 137, 148]
	Network lifetime	↑	[72]
	Processing/execution time	↓	[3, 6, 7, 9, 11, 12, 14, 51–55, 58, 61, 63, 64, 66, 67, 72–75, 80, 81, 83, 84, 92, 106, 108, 112, 115, 116, 121, 124, 125, 129, 131, 132, 135, 136, 142–145, 148, 160, 161, 163, 173, 189]
	Response time	↓	[7, 51, 52, 54, 58, 59, 61, 63, 68, 70–72, 77, 81, 83, 86, 88, 89, 92, 101, 105, 112, 121, 125, 128, 129, 141, 142, 146, 159, 173]
	Delay	↓	[30, 72, 81, 83, 86, 101, 106, 121, 143, 147, 148]
	CPU usage	↓	[57, 70, 83, 84, 86, 108, 109, 141, 148, 173]
	Memory usage	↓	[12, 30, 52, 57, 64, 65, 83, 86, 108, 109, 124, 148, 161, 173]
	Bandwidth usage	↓	[7, 59, 73, 80, 86, 101, 121, 137, 148]
	Throughput	↑	[7, 30, 51–53, 53, 55, 63–66, 66, 68, 70, 72, 74, 83, 89, 91, 92, 105, 120, 129, 141, 144]
	Latency	↓	[7, 54, 56, 59, 68, 70, 71, 73, 77, 83, 84, 86, 88, 90, 101, 105, 108, 109, 121, 143, 144, 148]
	Concurrency	↑	[68, 83]
Scalability			[6–8, 51–56, 59, 63, 66–68, 70, 72–76, 78, 83, 85, 86, 90, 101, 105–108, 111–113, 116, 117, 119, 121, 125, 129, 134, 141–144, 147, 159, 183]
Feasibility			[7–9, 30, 51, 53, 58, 59, 61, 63, 74, 77, 84, 105, 112, 118, 119, 121, 125, 131, 135, 143, 144]
Reliability			[11, 53, 55, 58, 59, 62, 63, 68, 70–73, 75, 77, 81–84, 86, 87, 89, 90, 105, 106, 121, 128, 129, 134, 137, 141, 143, 144, 147, 156]
Generality			[58, 65, 78, 89, 134]
Accuracy			[3, 6, 9, 11, 12, 30, 52, 52, 57–59, 59, 61, 62, 64, 65, 69, 71, 75–77, 77, 83–85, 85, 87, 101, 106, 108, 115–117, 119–125, 125, 127, 130–133, 133, 135, 136, 139, 141, 145, 158–161, 161]
Efficiency			[6, 8, 11, 12, 14, 30, 51, 52, 54–59, 61, 63–66, 69, 70, 72, 74–81, 83–88, 91, 92, 105, 106, 111–113, 115, 117, 119, 121, 122, 125, 127–129, 131, 133–135, 143, 144, 147, 160, 173, 189]
Applicability			[6, 11, 54, 56, 58, 63, 69, 70, 91, 108, 110, 135, 147, 159]
Confidentiality			[76, 112, 126]
Flexibility			[8, 11, 12, 14, 53, 56, 59, 67, 74, 75, 78, 80, 87, 108, 111–113, 116, 121, 128, 132, 134, 142, 157, 173]
Interoperability			[53, 54, 56, 59, 67, 105, 107, 121, 140]
Sustainability			[14, 86, 125, 162]
Security			[11, 14, 30, 54, 59, 68, 76, 82, 86, 88, 101, 107, 126, 134, 136, 141, 143, 144, 148, 156]
Usability			[59, 69, 75, 108, 119, 144]
Availability			[8, 11, 53, 55, 58, 59, 62, 63, 67, 68, 70, 72, 73, 75, 77, 81–84, 86, 89, 90, 105, 106, 111, 118, 121, 128, 129, 134, 137, 141–144, 147, 156, 183]
Robustness			[59, 69, 75, 108, 119, 144]
Accessibility			[53, 68, 90, 147]
Maintainability			[68, 76, 112, 121, 126]

Table 22 Classification of articles based on ISO 25010 software quality standard

Quality attributes	Article	#Articles
Functional Suitability	[6, 9, 11, 12, 52, 53, 57–59, 61, 62, 64, 69, 73, 75–77, 79, 83–85, 101, 106, 108, 111, 115–117, 119, 121–123, 125, 127, 128, 130, 131, 135, 136, 139, 141, 156, 159–161, 163]	46
Performance Efficiency	[6, 7, 9, 11, 12, 14, 30, 51–59, 61, 63, 65–70, 72–77, 80–84, 86–89, 91, 92, 101, 105, 106, 108, 109, 111, 112, 115, 116, 120, 121, 124, 125, 128, 129, 131–133, 135–137, 141–144, 146–148, 159–161, 163, 173, 189]	75
Compatibility	[53, 54, 56, 59, 105, 107, 121, 140]	8
Usability	[59, 69, 75, 108, 119, 144]	6
Reliability	[11, 53, 55, 58, 59, 62, 63, 68, 70, 72, 73, 75, 77, 81–84, 86, 87, 89, 90, 105, 106, 121, 128, 129, 134, 137, 141, 143, 144, 147, 156]	33
Security	[11, 14, 30, 54, 59, 68, 76, 82, 86, 88, 101, 107, 112, 126, 134, 136, 141, 143, 144, 148, 156]	21
Maintainability	[68, 76, 112, 121, 126]	5
Portability	[58, 78, 87, 89, 134]	5

deep learning, data mining, prediction mechanisms, etc., for BDA and to evaluate the data [198]. The articles used ML algorithms to perform various analyses in the steps of BDA. ML algorithm is an appropriate approach or tool for BDA; decision-making; meaningful, precise, and valuable information extraction; and detecting hidden patterns in big datasets [199, 200]. Utilizing the ML algorithms in BDA has advantages such as improving and optimizing BDM processes; heterogeneous big data analysis; sustainability; fault detection, prediction, and prevention; accurate and reliable real-time processing; resource management and reduction; and increased quality prediction, visual inspection, and productivity in IoT applications [83, 201]. These algorithms are divided into four types: supervised, semi-supervised, unsupervised, and reinforcement ML algorithms [53, 202]. Table 23 shows the categorization of articles based on BDA types. The most common tactics that the selected articles use for BDM in the IoT include classification (51 articles), simulation (38 articles), optimization (30 articles), and clustering (25 articles).

The reason for using more classification algorithms is that they help to categorize unstructured and high-volume data. Therefore, BDM in the IoT is faster and more efficient. Before classification begins, it must optimize the classification algorithm's inputs. Data reduction strategies extract optimal and required data from a large amount of data. These strategies include dimensionality reduction, numerosity reduction, and data compression. Some reviewed articles used Principal Components Analysis (PCA) to standardize, reduce the data redundancy and dimensionality, reduce the cost and processing time, and maintain the original data [69, 114, 118, 135, 136]. Also, the authors in [160] used the fuzzy C-means algorithm to reduce the amount of data. Feature selection methods improve classification accuracy and reduce the number of features in BDA. The collected data from IoT applications and monitoring systems are usually anomalous, and it is difficult to distinguish between the original data and the anomaly [201]. The anomaly and outlier data reduce the accuracy of the classification and prediction models. For instance, NRDD-DBSCAN [114], DBSCAN-based outlier detection [83], GA, and One-Class Support Tucker Machine (OCSTuM) [122, 124] are some of the high-robust, high-performance, and anti-noisy methods for anomaly detection that are presented in reviewed articles.

SVM is the most common method based on classification (10 articles) for BDM in the IoT in supervised classification. SVM is a non-parametric, memory-efficient, error-reduction classification method that performs well in theoretical analysis and real-world applications. It can model non-linear, complex, and real-world problems in high-dimensional feature space [2, 69, 203]. However, SVM is difficult to interpret, has a high computational cost,

Table 23 The classification of articles based on the big data analytic types in the IoT

Analytics Types	Taxonomy	Tactics Type			#Articles
Descriptive (31.82% articles)	Association rules	General association rules tactic [79, 106]			2
	Clustering	Fuzzy C-means algorithm [69, 71, 160]	Density-based spatial [83]	NRDD-DBSCAN [114]	25
		Tensortrain Network [161]	Mini batch K-means [135]	K-median [86]	
		K-means [56, 62, 75, 115, 131, 159]	Dynamic group clustering [135]	K-mediods [119]	
		Incremental clustering algorithm [119]	Community detection [125]	Graph-based clustering [125]	
		High-order Possibilistic C-means algorithm [161]	Gustafson-Kessel [69]	Mongo clustering [134]	
		General/new clustering tactic [12, 111, 123, 144, 189]	DBSCAN-based clustering [83]	Single-linkage clustering [133]	
			Parallel military dog based algorithm [145]	MDL with hierarchical clustering [65]	
	Pattern mining	Frequent pattern growth algorithm [56]			1
	Anomaly detection	GA and OCSTuM [122]		NRDD-DBSCAN [114]	8
		ADASYN [124]		Persistent querying of the infinite streams [105]	
		General anomaly detection [84, 133]		Multicriteria temporal graphs [214]	
		DBSCAN-based outlier detection [83]			
	Principal Component Analysis (PCA)	Divide and Conquer PCA [78]			6
		General PCA tactic [69, 114, 118, 135, 136]			
Diagnostic (1.82% articles)	Bayesian analyze	General Bayesian analysis tactic [84, 115]			2
Predictive (57.27% articles)	Classification	SVM		KNN [136]	51
		Linear kernel SVM [64]		Stability-based K-means [106]	
		General SVM tactic [69, 85, 90, 108, 111, 131, 136]		Bag of words technique [127]	
		LibSVM libraries [52, 148]		OCSTuM [122]	
		An unsupervised form of classification [56]		Spark MLlib [53, 107, 109]	
		CDNN [130, 163]		SVM classifier [85]	
		Decision tree		J48 algorithm [76]	
		Hoeffding tree [115]		Information gain technique [83, 115]	
		General decision tree tactic [14, 59, 70, 111, 136]		NB [30, 84, 115]	
		Classification and Regression Tree (CART) [58]		Wrapper feature selection [115]	
		Resource description framework [140]		Gaussian NB [136]	
		Generative adversarial network [138]		Unsupervised classification [13]	
		LSTM [124]		ML classifier [60, 64, 127]	
		Fuzzy C-means [71]		CEP [121]	
		Random forests		Reptree [58, 61, 84]	

Table 23 (continued)

Analytics Types	Taxonomy	Tactics Type			#Articles
		MLlib framework [108]		RNN [133]	
		General random forests tactic [58, 59, 83, 111, 139]		Logistic regression [109]	
		Deep NN based marine predator [120]			
		General/new classification tactic [7, 12, 51, 68, 70, 74, 75, 87, 89, 111, 123, 126, 128, 141]			
	Regression	Based on radial basis functions network [69]		Random forests [58]	8
		Linear regression [59, 70, 79]		CART [58]	
		LWLR [73]		Logistic regression [109, 141]	
	Time series analysis	General time series analysis tactic [113]		Holt-Winter's forecasting [75]	3
		Symbolic aggregate approximation algorithm [159]			
	Neural Networks	Radial basis functions network [69]	Two-layer ANN [136]	Backpropagation [6, 9]	17
		Optimized RNN [133]	LSTM [108]	Deep NN [101, 162]	
		High-order backpropagation algorithm [163]	Convolutional NN [163]	Temporal RNN [71]	
		General NN tactic [111, 131, 158]	MLP [59, 69, 84]	BCN [87]	
			CDNN [130]		
	Deep learning	Deep-learning-based greedy algorithm [116]		EDL [87]	4
		CDNN [130, 163]			
	AdaBOOST	General AdaBOOST tactic [111]			1
	Feature selection	Fuzzy rough [84]		Divide and Conquer PCA [78]	12
		Pearson correlation analysis [59]		GA [122]	
		EHO [64]		Correlation Feature selection (CFS) [115]	
		ABC [52]		ML-based [111]	
		Wrapper feature selection [115]		Quantum EHO [138]	
		Fuzzy rules and valuable features selection [133]		High-variance feature removal [118]	
		Modified relief technique [120]			
Prescriptive (50.91% articles)	Simulation	[3, 8, 9, 11, 14, 30, 52, 55, 57, 62–66, 71, 76–78, 81–83, 85, 86, 101, 105, 106, 109, 115, 116, 124, 128, 131, 137, 140, 143, 144, 146, 156]			38
	Optimization	PSO [6, 115, 173]		Hybrid algorithms [11]	30
		Improved dragonfly algorithm [139]		ABC [52]	
		Sequential minimal optimization algorithms [84]		Dynamic group optimization [135]	
		Stochastic gradient descent algorithm [141]		NN [6, 9, 111, 131, 158, 163]	
		Mixed integer linear programming model [147]		Incremental clustering algorithm [119]	
		Grid search methods [85]		EHO [64]	
		Optimized cluster storage method [144]		Heuristic approach [136]	
		Search engine optimization techniques [72]		Multi-objective optimization [146]	

Table 23 (continued)

Analytics Types	Taxonomy	Tactics Type		#Articles
		Bloom filter based optimization [117]	Ant Colony Algorithm [143]	
		GA [69, 86, 115, 122, 146, 173]	Parallel military dog based algorithm [145]	
		OSSO [124]		
		Optimized RNN [133]		

and is not scalable [204]. In unsupervised classification, the k-means clustering algorithm is the most common strategy (6 articles). The standard k-mean clustering algorithm is a simple partitioning method that works well for small and structured datasets. It is sensitive to the number of clusters, initial input, and noise data. The standard k-means clustering must be modified to be used in BDA. Some research focuses on the MapReduce/Spark implementation of traditional k-means clustering that improves the accuracy and reduces the time complexity [205]. Also, articles used the k-means clustering algorithm to predict floods [75], security monitoring [136], energy management and improve the prediction accuracy [56, 131], the data access and resource utilization [144] in IoT. Association rules are an unsupervised learning approach used to discover interesting and hidden relationships and correlations between variables and objects in large databases and for data modeling in IoT [79]. Association rule mining uses various algorithms to identify frequent item sets, such as the apriori algorithm, FP growth algorithm, and maximal frequent itemset algorithm [79, 106]. Neural networks (NN) perform big data processing and analysis efficiently. NN has self-learning ability and plays a significant role in BDA in IoT. NN is used for classification, big data mining, hidden pattern recognition, correlation recognition in big data raw, and decision-making in IoT applications. There are several different kinds of neural network algorithms, including LSTM [108], radial basis functions network [69], Deep NN [101, 162], convolutional NN [163], etc.

Deep learning is a modern machine learning model that employs supervised or unsupervised methods to learn and extract multiple-level, high-level, and hierarchical features for big data classification tasks and pattern recognition [163, 206]. Deep learning is a BDA tool that can speed up big data decision-making and feature extraction, improve the extracted information QoE level, resolve security issues, data dimensionality, and unlabeled and un-categorized big data processing in IoT applications [116, 207]. In the reviewed articles, deep learning methods are used for human activity recognition [87], flood detection [130], smart cities [116], and feature learning on big data in the IoT [163]. Optimization refers to selecting the best solution from a set of alternatives by minimizing or maximizing a specified objective function [208]. Bio-inspired algorithms are stochastic search techniques used by many researchers to solve optimization problems in BDM processes in the IoT, including data ingestion, processing, analytics, and virtualization [209]. The features of these algorithms are good applicability, simplicity, robustness, flexibility, self-organization, and the possibility of dealing with real-world problems [210]. There are different types of categories for these algorithms in various articles. For instance, in [211], these algorithms are categorized into six categories: local search-based and global search-based; single-solution based and population-based; memory-based and memory-less; greedy and iterative; parallel; and nature-inspired and hybridized. In the reviewed articles, GA and NN are used more for BDM in the IoT (6 articles). GA has been used for feature extraction and selection, outlier detection, scheduling, optimizing energy consumption, reducing execution time and delay, and optimizing the predictive model in IoT applications [69, 86, 115, 122, 146, 173].

4 Open issues and challenge

This section offers a variety of vital issues and challenges that require future work. IoT faces many challenges and open issues, including security, privacy, hardware, heterogeneity, data analysis, and virtualization challenges. IoT devices produce big data that must be monitored and managed using particular data patterns. For efficient decision-making, BDA in the IoT is applied to large datasets to reveal unseen patterns and correlations. So the key challenge in big data in the IoT is analyzing that data for knowledge discovery and virtualization. Various types of research have presented different categories for challenges and open issues for BDM in the IoT. Romero et al. [212] divided challenges into principal worries, security and monitoring, technological development, standardization, and privacy. Santana et al. [213] divided challenges into privacy, data management, heterogeneity, energy management, communication, scalability, security, lack of testbed, city models, and platform maintenance. Ahmed et al. [27] divided challenges into four categories: diversity, security, data provenance, data management, and data governance and regulation. This study divides challenges into BDM in the IoT and quality attributes challenges.

4.1 Big data management in the IoT challenges

In many reviewed articles, IoT big data management depends on centralized centers, including cloud-based servers, and has technical limitations. These architectures are platform-centric and have costly customized access mechanisms. A centralized architecture can have a single point of failure, which is very inefficient in terms of scalability and reliability. Also, in these architectures, unauthorized access to the server might easily result in the modification, leak, or manipulation of critical data [215]. In some research, authors used blockchain technology to overcome these problems [215, 216]. But this technology has some challenges. For example, blockchain platforms can consume IoT devices' computational resources extensively. During the review in Sect. 3.1, the process of BDM in the IoT includes data collection, communication, data ingestion, data storage, processing and analysis, and post-processing, each of which faces a variety of challenges and problems. This section examines the challenges involved in each of these steps.

4.1.1 Data collection

Big data in the IoT is generated from different, distributed, and multisource heterogeneous unsupervised domain [217, 218]. Collecting this large amount of diverse data faces challenges such as energy consumption, limited battery life in sensors and other data collection devices, different hardware and operating systems, multiple and disparate resources, and combining them. It can be difficult to obtain complete, accurate, and maintain quality data. IoT and WSN encompass a large number of distributed mobile nodes. Mobile nodes [219] must increase the amount of data collected while minimizing the power consumption of both the mobile node and IoT devices. Therefore, the main challenge is mobile data collection management, determining and planning mobile sink trajectories for collecting data from nodes. Most existing mobile data collection approaches are static and only find a solution for a scenario with fixed parameters [220]. These solutions do not consider the change in the amount of data generated by the IoT nodes or devices when an IoT device can move from one situation to another. For future work, we propose using AI techniques, including ML or deep learning, for intelligent management of mobile data collection.

4.1.2 Communication

Transferring data from different sources to the data processing and analysis stage is one of the steps in BDM in the IoT. Communication protocols and technologies must share data at high speeds and on time. The connectivity challenges include interoperability, bandwidth, reducing traffic, energy consumption, security, network, transport protocols, delivery of services, network congestion, and communication cost. Another connectivity challenge is nodes accessing other nodes' information under different network topologies with different channel fading [221]. Concerning advances in mobile information infrastructure, integration of the 6G technologies, mobile satellite communications, and AI can increase frequency band, network speed, and network coverage and improve the number of connections [222]. Different approaches are proposed for data transmission optimizing and overcoming these limitations, such as parsimonious/compressive sensing [223, 224]. Compressive sensing technology is a theory of acquiring and compressing signals that use the sparsity behavior of natural signals at the sensing stage to minimize power consummation and data dimensionality reduction [225]. In compressive sensing technology, the collected data from different sensors are first compressed and then transmitted. Therefore, the complexity is transferred to the receiver side from the sensors, which are usually resource-constrained and self-powered [226]. For future work, we propose combining compressive sensing with AI technologies to present a lightweight, real-time, and dynamic compressive sensing method for overcoming the communication challenges in BDM in the IoT.

4.1.3 Data ingestion

Big data in the IoT have various features such as: enormous, high-speed, heterogeneity of data formats, complexity, different data resolutions, abnormal and incorrect, ambiguity, unbalanced, massive redundancy, multidimensional, granularity, continuously, inconsistencies, probabilistic, sparse, sequential, dynamical, timeliness, non-randomly distributed, and misplaced [56, 63, 89, 117, 119, 125, 135, 137, 173, 227]. Each data ingestion step discussed in Sect. 3.1.3 has challenges. These issues are anomaly detection, missing data, outlier detection, feature selection/extraction, dimensionality reduction, redundancy, standardization, rule discovery, computational cost, and normalization that different mechanisms use for these challenges. Missing data could lead to the loss of a large amount of valuable and reliable information and bad decision-making. Many articles utilize the delete, ignore, mean/median value, or constant global methods for handling missing data. These dangerous methods may yield biased and untrustworthy results [228]. Therefore, adding new techniques by considering more efficiency, high accuracy, minimal computational complexity, and less time consumption is interesting in the future. For this purpose, we can use ML and nature-inspired optimization algorithms or a combination thereof. The parallel technology has made data ingestion and processing more efficient in recent years, and it saves space and time by eliminating the need to decompress data [229]. Also, BDA types in the IoT are used in this stage, which is discussed in Sect. 3.4. Each of these methods has challenges. For example, clustering has challenges such as real-time clustering, local optima, determining the number of clusters, updating the clustering centers, and determining the initial clustering centers. ANN faces many issues, including how to determine the number of layers, the training, and test samples, the number of nodes, choosing an operable objective function, and how to improve the training speed of the network in a big data environment. Various articles solve these problems using meta-heuristic algorithms. However, these algorithms cannot handle big IoT data sets within the specified time due to high computation costs, limited memory, and processing units, and premature convergence [145, 230]. For future work, we propose using new optimization meta-heuristic algorithms and AI methods based on these techniques by utilizing the strengths of MapReduce and Apache Spark.

4.1.4 Data storage

Data storage is another major challenge in BDM in the IoT. The big data storage mechanisms in the IoT were discussed in Sect. 3.1.4. The challenges in this regard can be categorized as IoT-based big data storage systems in cloud computing and complex environments such as industry 4.0 applications and data storage architecture. The main data storage challenges are IoT data replication and consistency management. Many researchers have proposed strategies for determining the best location for copy storage in geo-distributed storage systems based on cloud and fog computing. But many of them, due to the geographical distance between distributed storage systems, cannot handle the problems of high data access latencies and replica synchronization costs [231]. Also, data consistency management strategies must manage the massive amounts of data with different data consistency requirements and system heterogeneity.

4.1.5 *Processing* and analysis

The big data processing and analysis in BDM in the IoT has different challenges, including task scheduling, real-time data analysis, developing the IoT data analysis infrastructure, data management in the cloud-IoT environments, and query optimization. The authors used data mining and AI algorithms to overcome these challenges. The challenges of using AI technologies for data analytics in the IoT are to balance the computational costs (or response time) and improve the accuracy of the prediction and analysis results [232]. Also, many multi-objective optimization problems have more than three objective functions, which present challenges, including the diversity and convergence speed of the algorithm [152]. However, determining an algorithm to process a dynamic IoT dataset based on some application-specific goals for better accuracy remains a challenge. Also, most current methods cannot meet user demands for the fundamental features of cloud-IoT environments, including heterogeneity, dynamism, reliability, flexibility, responsiveness, and elasticity. For future work, we propose studies of various optimization algorithms, including metaheuristic algorithms (many-objective) and ML algorithms, and combined versions of these algorithms for big data processing and analysis in the IoT. Regarding the limitations of wireless nodes (low power and computational) and cloud servers (high latency, privacy, performance bottleneck, context unawareness, etc.) for processing and analysis computing tasks, using mobile edge or fog computing to overcome these problems is helpful.

4.1.6 Post-processing

Providing insight from processed and analyzed data in the IoT requires selecting appropriate visualization techniques. Most of the reviewed methods use simulator tools such as CloudSim [143, 173], TRNSYS [131], Cooja [82], and Extend-Sim [8] for evaluation. Additional studies are needed to evaluate the mentioned approaches in real-world systems and datasets.

4.2 QoS management

QoS is one of the critical factors in BDM in the IoT and needs research, management, and optimization (discussed in Sect. 3.3). The reviewed articles used these parameters and metrics for evaluation. No article considers these parameters thoroughly for its proposed architecture. Therefore, it is exciting to compare various architectures by considering the different QoS parameters and quality attributes in the future. Security, privacy, and trust are critical issues in IoT BDA that most reviewed articles did not address, and the proposed architectures or frameworks did not involve the data perception layer. The security frame generally consists of confidentiality, integrity, authentication, non-repudiation, availability, and privacy [233]. We concede that no comprehensive and highly secure scheme or platform for all types of data collection, analysis, and sharing meets all security requirements. The other main challenges are integrating privacy protection methods with data sharing platforms and selecting the best privacy protection algorithms to use during data processing [172]. Therefore, it is suggested for the future to utilize cryptographic mechanisms in different layers of architectures or frameworks, add a data perception layer, and develop security protocols specifically for IoT devices because of their heterogeneity and resource limitations.

The blockchain framework is widely used in IoT to improve protection, trust, reputation, management, control, and security. The blockchain framework provides decentralized security, authentication rules, and privacy for IoT devices. However, there are major challenges, such as high energy consumption, delay, and computational overhead, because of the resource constraints in IoT devices. Many types of research have been suggested as solutions to these problems. For instance, Corradini et al. [234] proposed a two-tier Blockchain framework for increasing the security and autonomy of smart objects in the IoT by implementing a trust-based protection mechanism. The tiers of this framework are a point-to-point local tier and a community-oriented global tier. Pincheira et al. [235] proposed a cost-effective blockchain-based architecture for ensuring data integrity, auditability, and traceability and increasing trust and trustworthiness in IoT devices. This architecture has four components: the cloud module, mobile app, connected tool, and blockchain module. Tchagna Kouanou et al. [236] proposed a 4-layer blockchain-based architecture to secure data in the IoT to increase security, integrity, scalability, flexibility, and throughput. The layers of this architecture are tokens, smart contracts, blockchain, and peers. In future research, we suggest using AI techniques and a lightweight blockchain framework to increase protection, trust, reputation, and security in the IoT.

Trust and reputation management are vital issues in the SIoT and MIoT scenarios. In [237], the authors defined trust and reputation in the MIoT as the trust of an instance in another one of the same IoT; the trust of an object in another one of the MIoT; the reputation of an instance in an IoT; the reputation of an object in a MIoT; the reputation of an IoT in a MIoT; the trust of an IoT in another IoT; and the trust of an object in an IoT. Security in the SIoT aims to differentiate between secure and malicious things and increase the safety and protection of SIoT networks [185]. Investigating trust and reputation in SIoT and MIoT has many benefits, such as identifying, isolating, managing malicious objects, supporting collaboration, and identifying and evaluating the objects' QoS parameters. Also, the lack of trust and reputation management in SIoT and MIoT causes problems such as loss of accessibility, privacy, and security [237]. To overcome these issues, we suggest utilizing trust and reputation with AI methods to develop detection techniques for anomalous and malicious behaviors of things in the MIoT and SIoT in future works.

5 Conclusion

This paper presented a systematic review of the BDM mechanisms in the IoT. First, we discussed the advantages and disadvantages of some systematic and review articles about BDM in the IoT and then explained the purpose of this paper. Then, the research methodology and details of 110 selected articles were presented. These articles were divided into four main categories, including BDM processes, big BDM architectures/frameworks, quality attributes, and data analytics types in IoT. Some of these categories have been divided into some subcategories: BDM process in IoT was divided into data collection, communication, data ingestion, data storage, processing and analysis, and post-processing; big data architectures/frameworks in the IoT were divided into BDM architectures/frameworks in the IoT-based applications and BDM architectures/frameworks in the IoT paradigms; big data analytics-types were divided into the descriptive, diagnostic, predictive, and prescriptive analysis; and big data storage systems in the IoT were divided into relational databases, NoSQL databases, DFS, and cloud/edge/fog/mist storage. Also, the advantages and disadvantages of each of the BDM mechanisms in the IoT were discussed. The tools and platforms used for BDM in the IoT in the articles were reviewed and compared based on criteria. The most common type of analysis that articles use is predictive analysis, with 57.27%, which uses ML algorithms. The classification, optimization, and clustering algorithms are the most widely used for big data analysis in the IoT. Some articles present architectures mostly in IoT-based healthcare, with 33.33%, and IoT-based smart cities, with 22.22%. These architectures have two to eight layers, each performing a set of functions. In the review of qualitative characteristics, we observed that most articles evaluated their evaluations based on criteria, including performance, efficiency, accuracy, and scalability. Meanwhile, some features are less used, including confidentiality, sustainability, accessibility, portability, generality, and maintainability. The NoSQL database and DFS are used more to store data than other databases. The BDM process in the IoT uses different algorithms and tools with various features. Various programming languages and operating systems are used to evaluate and implement the proposed mechanisms. The Java and python programming languages and the UBUNTU operating system are used more.

This paper tries to review the BDM mechanisms in the IoT. Specifically, it considers studies published in high-quality international journals. The most recent works on BDM mechanisms in the IoT have been compared and analyzed in this paper. We hope that this study will be helpful for the next generation of studies for developing BDM mechanisms in real-complex environments.

References

1. Cao, B., Zhang, Y., Zhao, J., Liu, X., Skonieczny, Ł., & Lv, Z. (2021). Recommendation based on large-scale many-objective optimization for the intelligent internet of things system. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2021.3104661
2. Hou, R., Kong, Y., Cai, B., & Liu, H. (2020). Unstructured big data analysis algorithm and simulation of internet of things based on machine learning. *Neural Computing and Applications, 32*, 5399–5407.
3. Kumar, M., Kumar, S., & Kashyap, P. K. (2021). Towards data mining in IoT cloud computing networks: Collaborative filtering based recommended system. *Journal of Discrete Mathematical Sciences and Cryptography, 24*, 1309–1326.
4. Cao, B., Zhao, J., Lv, Z., & Yang, P. (2020). Diversified personalized recommendation optimization based on mobile data. *IEEE Transactions on Intelligent Transportation Systems, 22*, 2133–2139.
5. Sanislav, T., Mois, G. D., Zeadally, S., & Folea, S. C. (2021). Energy harvesting techniques for internet of things (IoT). *IEEE Access, 9*, 39530–39549.
6. Zhou, H., Sun, G., Fu, S., Liu, J., Zhou, X., & Zhou, J. (2019). A Big data mining approach of PSO-based BP Neural network for financial risk management with IoT. *IEEE Access, 7*, 154035–154043.
7. Tang, B., Chen, Z., Hefferman, G., Pei, S., Wei, T., He, H., et al. (2017). Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Transactions on Industrial informatics, 13*, 2140–2150.
8. Jiang, W. (2019). An intelligent supply chain information collaboration model based on internet of things and big data. *IEEE Access, 7*, 58324–58335.
9. Xiao, S., Yu, H., Wu, Y., Peng, Z., & Zhang, Y. (2017). Self-evolving trading strategy integrating internet of things and big data. *IEEE Internet of Things Journal, 5*, 2518–2525.
10. Sowe, S. K., Kimata, T., Dong, M., & Zettsu K. (2014). Managing heterogeneous sensor data on a big data platform: IoT services for data-intensive science. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, Vasteras, Sweden, pp. 295-300
11. Nie, X., Fan, T., Wang, B., Li, Z., Shankar, A., & Manickam, A. (2020). Big data analytics and IoT in operation safety management in under water management. *Computer Communications, 154*, 188–196.
12. Liu, H., & Liu, X. (2019). A novel research on the influence of enterprise culture on internal control in big data and internet of things. *Mobile Networks and Applications, 24*, 365–374.
13. Piccialli, F., Benedusi, P., Carratore, L., & Colecchia, G. (2020). An IoT data analytics approach for cultural heritage. *Personal and Ubiquitous Computing*. https://doi.org/10.1007/s00779-019-01323-z
14. Liu, C., Feng, Y., Lin, D., Wu, L., & Guo, M. (2020). Iot based laundry services: an application of big data analytics, intelligent logistics management, and machine learning techniques. *International Journal of Production Research*. https://doi.org/10.1080/00207543.2019.1677961
15. Wang, J., Wu, Y., Yen, N., Guo, S., & Cheng, Z. (2016). Big data analytics for emergency communication networks: A survey. *IEEE Communications Surveys & Tutorials, 18*, 1758–1778.
16. Jahanbakht, M., Xiang, W., Hanzo, L., & Azghadi, M. R. (2020) Internet of underwater things and big marine data analytics–a comprehensive survey. *arXiv preprint* arXiv:2012.06712.
17. Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., & Markakis, E. K. (2020). A survey on the internet of things (IoT) forensics: Challenges, approaches, and open issues. *IEEE Communications Surveys & Tutorials, 22*, 1191–1221.
18. Aldalahmeh, S. A., & Ciuonzo, D. (2022). Distributed detection fusion in clustered sensor networks over multiple access fading channels. *IEEE Transactions on Signal and Information Processing over Networks, 8*, 317–329.

19. Rajavel, R., Ravichandran, S. K., Harimoorthy, K., Nagappan, P., & Gobichettipalayam, K. R. (2022). IoT-based smart healthcare video surveillance system using edge computing. *Journal of Ambient Intelligence and Humanized Computing, 13*, 3195–3207.
20. Shahid, H., Shah, M. A., Almogren, A., Khattak, H. A., Din, I. U., Kumar, N., et al. (2021). Machine learning-based mist computing enabled internet of battlefield things. *ACM Transactions on Internet Technology (TOIT), 21*, 1–26.
21. Thomas, D., Orgun, M., Hitchens, M., Shankaran, R., Mukhopadhyay, S. C., & Ni, W. (2020). A graph-based fault-tolerant approach to modeling QoS for IoT-based surveillance applications. *IEEE Internet of Things Journal, 8*, 3587–3604.
22. S. Vahdat (2020) The role of IT-based technologies on the management of human resources in the COVID-19 era. *Kybernetes*.
23. Hassan, M., Awan, F. M., Naz, A., deAndrés-Galiana, E. J., Alvarez, O., Cernea, A., et al. (2022). Innovations in genomics and big data analytics for personalized medicine and health care: A review. *International Journal of Molecular Sciences, 23*, 4645.
24. Honar Pajooh, H., Rashid, M. A., Alam, F., & Demidenko, S. (2021). IoT big data provenance scheme using blockchain on Hadoop ecosystem. *Journal of Big Data, 8*, 1–26.
25. Priyadarshini, S. B. B., Bhusan Bagjadab, A., & Mishra B. K. (2019). The role of IoT and big data in modern technological arena: A comprehensive study. In *Internet of Things and Big Data Analytics for Smart Generation.* Springer, pp. 13–25.
26. Zheng, W., Yin, L., Chen, X., Ma, Z., Liu, S., & Yang, B. (2021). Knowledge base graph embedding module design for Visual question answering model. *Pattern Recognition, 120*, 108153.
27. Ahmed, E., Yaqoob, I., Hashem, I. A. T., Khan, I., Ahmed, A. I. A., Imran, M., et al. (2017). The role of big data analytics in internet of things. *Computer Networks, 129*, 459–471.
28. Singh, S., & Yassine, A. (2018). IoT big data analytics with fog computing for household energy management in smart grids. In *International Conference on Smart Grid and Internet of Things*. pp. 13–22.
29. Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa A., et al. (2017). Big IoT data analytics: architecture, opportunities, and open research challenges. *ieee access*, 5, 5247–5261.
30. Li, C. (2020). Information processing in internet of things using big data analytics. *Computer Communications, 160*, 718–729.
31. Kwon, O., Lee, N., & Shin, B. (2014). Data quality management, data usage experience and acquisition intention of big data analytics. *International journal of information management, 34*, 387–394.
32. Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management, 35*, 137–144.
33. Ahmed, M., Choudhury, S., & Al-Turjman, F. (2019). Big data analytics for intelligent internet of things. In *Artificial Intelligence in IoT*. Springer, pp. 107–127.
34. Urrehman, M. H., Ahmed, E., Yaqoob, I., Hashem, I. A. T., Imran, M., & Ahmad, S. (2018). Big data analytics in industrial IoT using a concentric computing model. *IEEE Communications Magazine, 56*, 37–43.
35. Constante Nicolalde, F., Silva, F., Herrera, B., & Pereira, A. (2018). Big data analytics in IOT: challenges, open research issues and tools. In *World conference on information systems and technologies*, pp. 775–788.
36. Talebkhah, M., Sali, A., Marjani, M., Gordan, M., Hashim, S. J., & Rokhani, F. Z. (2021). IoT and big data applications in smart cities: Recent advances, challenges, and critical issues. *IEEE Access, 9*, 55465–55484.
37. Bansal, M., Chana, I., & Clarke, S. (2020). A survey on iot big data: Current status, 13 v's challenges, and future directions. *ACM Computing Surveys (CSUR), 53*, 1–59.
38. Simmhan, Y., & Perera, S. (2016). Big data analytics platforms for real-time applications in IoT. In *Big data analytics*. Springer, pp. 115–135.
39. Shoumy, N. J., Ang, L.-M., Seng, K. P., Rahaman, D. M., & Zia, T. (2020). Multimodal big data affective analytics: A comprehensive survey using text, audio, visual and physiological signals. *Journal of Network and Computer Applications, 149*, 102447.
40. Ge, M., Bangui, H., & Buhnova, B. (2018). Big data for internet of things: A survey. *Future Generation Computer Systems, 87*, 601–614.
41. Siow, E., Tiropanis, T., & Hall, W. (2018). Analytics for the internet of things: A survey. *ACM Computing Surveys (CSUR), 51*, 1–36.
42. Fawzy, D., Moussa, S. M., & Badr, N. L. (2022). The internet of things and architectures of big data analytics: Challenges of intersection at different domains. *IEEE Access, 10*, 4969–4992.
43. Zhong, Y., Chen, L., Dan, C., & Rezaeipanah, A. (2022). A systematic survey of data mining and big data analysis in internet of things. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-022-04594-1
44. Hajjaji, Y., Boulila, W., Farah, I. R., Romdhani, I., & Hussain, A. (2021). Big data and IoT-based applications in smart environments: A systematic review. *Computer Science Review, 39*, 100318.
45. Ahmadova, U., Mustafayev, M., Kiani Kalejahi, B., Saeedvand, S., & Rahmani, A. M. (2021). Big data applications on the internet of things: A systematic literature review. *International Journal of Communication Systems, 34*, e5004.
46. Doewes, R. I., Gharibian, G., Zadeh, F. A., Zaman, B. A., Vahdat, S., & Akhavan-Sigari, R. (2022). An updated systematic review on the effects of aerobic exercise on human blood lipid profile. *Current Problems in Cardiology*. https://doi.org/10.1016/j.cpcardiol.2022.101108
47. Zadeh, F. A., Bokov, D. O., Yasin, G., Vahdat, S., & Abbasalizad-Farhangi, M. (2021). Central obesity accelerates leukocyte telomere length (LTL) shortening in apparently healthy adults: A systematic review and meta-analysis. *Critical Reviews in Food Science and Nutrition*. https://doi.org/10.1080/10408398.2021.1971155
48. Esmailiyan, M., Amerizadeh, A., Vahdat, S., Ghodsi, M., Doewes, R. I., & Sundram, Y. (2021). Effect of different types of aerobic exercise on individuals with and without hypertension: An updated systematic review. *Current Problems in Cardiology*. https://doi.org/10.1016/j.cpcardiol.2021.101034
49. Vahdat, S., & Shahidi, S. (2020). D-dimer levels in chronic kidney illness: a comprehensive and systematic literature review. *Proceedings of the National Academy of Sciences, India Section b: Biological Sciences*. https://doi.org/10.1007/s40011-020-01172-4
50. Zhou, D., Yan, Z., Fu, Y., & Yao, Z. (2018). A survey on network data collection. *Journal of Network and Computer Applications, 116*, 9–23.
51. Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks, 101*, 63–80.
52. Ahmad, A., Khan, M., Paul, A., Din, S., Rathore, M. M., Jeon, G., et al. (2018). Toward modeling and optimization of features selection in big data based social Internet of Things. *Future Generation Computer Systems, 82*, 715–726.

53. Shah, S. A., Seker, D. Z., Rathore, M. M., Hameed, S., Yahia, S. B., & Draheim, D. (2019). Towards disaster resilient smart cities: Can internet of things and big data analytics be the game changers? *IEEE Access, 7*, 91885–91903.
54. Celesti, A., & Fazio, M. (2019). A framework for real time end to end monitoring and big data oriented management of smart environments. *Journal of Parallel and Distributed Computing, 132*, 262–273.
55. Silva, B. N., Khan, M., & Han, K. (2017). Integration of big data analytics embedded smart city architecture with RESTful web of things for efficient service provision and energy management. *Future generation computer systems*. https://doi.org/10.1016/j.future.2017.06.024
56. Yassine, A., Singh, S., Hossain, M. S., & Muhammad, G. (2019). IoT big data analytics for smart homes with fog and cloud computing. *Future Generation Computer Systems, 91*, 563–573.
57. Khan, M., Han, K., & Karthik, S. (2018). Designing smart control systems based on internet of things and big data analytics. *Wireless Personal Communications, 99*, 1683–1697.
58. Rathore, M. M., Paul, A., Ahmad, A., Anisetti, M., & Jeon, G. (2017). Hadoop-based intelligent care system (HICS) analytical approach for big data in IoT. *ACM Transactions on Internet Technology (TOIT), 18*, 1–24.
59. Yacchirema, D. C., Sarabia-Jácome, D., Palau, C. E., & Esteve, M. (2018). A smart system for sleep monitoring by integrating IoT with big data analytics. *IEEE Access, 6*, 35988–36001.
60. Ma, Y., Wang, Y., Yang, J., Miao, Y., & Li, W. (2016). Big health application system based on health internet of things and big data. *IEEE Access, 5*, 7885–7897.
61. Rathore, M. M., Ahmad, A., Paul, A., Wan, J., & Zhang, D. (2016). Real-time medical emergency response system: Exploiting IoT and big data for public health. *Journal of medical systems, 40*, 283.
62. Zhou, Q., Zhang, Z., & Wang, Y. (2019). WIT120 data mining technology based on internet of things. *Health Care Management Science*. https://doi.org/10.1007/s10729-019-09497-x
63. Silva, B. N., Khan, M., Jung, C., Seo, J., Muhammad, D., Han, J., et al. (2018). Urban planning and smart city decision management empowered by real-time data processing using big data analytics. *Sensors, 18*, 2994.
64. Lakshmanaprabu, S., Shankar, K., Khanna, A., Gupta, D., Rodrigues, J. J., Pinheiro, P. R., et al. (2018). Effective features to classify big data using social internet of things. *IEEE access, 6*, 24196–24204.
65. Al-Qurabat, A. K. M., Mohammed, Z. A., & Hussein, Z. J. (2021). Data traffic management based on compression and MDL techniques for smart agriculture in IoT. *Wireless Personal Communications, 120*, 2227–2258.
66. Ahmad, A., Babar, M., Din, S., Khalid, S., Ullah, M. M., Paul, A., et al. (2019). Socio-cyber network: The potential of cyber-physical system to define human behaviors using big data analytics. *Future generation computer systems, 92*, 868–878.
67. Floris, A., Porcu, S., Atzori, L., & Girau, R. (2022). A Social IoT-based platform for the deployment of a smart parking solution. *Computer Networks, 205*, 108756.
68. Al-Ali, A.-R., Zualkernan, I. A., Rashid, M., Gupta, R., & AliKarar, M. (2017). A smart home energy management system using IoT and big data analytics approach. *IEEE Transactions on Consumer Electronics, 63*, 426–434.
69. Moreno, M. V., Terroso-Sáenz, F., González-Vidal, A., Valdés-Vela, M., Skarmeta, A. F., Zamora, M. A., et al. (2016). Applicability of big data techniques to smart cities deployments. *IEEE Transactions on Industrial Informatics, 13*, 800–809.
70. Nasiri, H., Nasehi, S., & Goudarzi, M. (2019). Evaluation of distributed stream processing frameworks for IoT applications in smart cities. *Journal of Big Data, 6*, 52.
71. Ahanger, T. A., Tariq, U., Nusir, M., Aldaej, A., Ullah, I., & Sulman, A. (2022). A novel IoT–fog–cloud-based healthcare system for monitoring and predicting COVID-19 outspread. *The Journal of Supercomputing, 78*, 1783–1806.
72. Rani, S., & Chauhdary, S. H. (2018). A novel framework and enhanced QoS big data protocol for smart city applications. *Sensors, 18*, 3980.
73. Lu, Z., Wang, N., Wu, J., & Qiu, M. (2018). IoTDeM: An IoT big data-oriented MapReduce performance prediction extended model in multiple edge clouds. *Journal of Parallel and Distributed Computing, 118*, 316–327.
74. Rathore, M. M., Paul, A., Hong, W.-H., Seo, H., Awan, I., & Saeed, S. (2018). Exploiting IoT and big data analytics: Defining smart digital city using real-time urban data. *Sustainable cities and society, 40*, 600–610.
75. Sood, S. K., Sandhu, R., Singla, K., & Chang, V. (2018). IoT, big data and HPC based smart flood management framework. *Sustainable Computing: Informatics and Systems, 20*, 102–117.
76. Machorro-Cano, I., Alor-Hernández, G., Paredes-Valverde, M. A., Rodríguez-Mazahua, L., Sánchez-Cervantes, J. L., & Olmedo-Aguirre, J. O. (2020). HEMS-IoT: A big data and machine learning-based smart home system for energy saving. *Energies, 13*, 1097.
77. Raptis, T. P., Passarella, A., & Conti, M. (2018). Performance analysis of latency-aware data management in industrial IoT networks. *Sensors, 18*, 2611.
78. Seng, K. P., & Ang, L.-M. (2018). A big data layered architecture and functional units for the multimedia Internet of Things. *IEEE Transactions on Multi-Scale Computing Systems, 4*, 500–512.
79. Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., & Nillaor, P. (2019). IoT and agriculture data analysis for smart farm. *Computers and electronics in agriculture, 156*, 467–474.
80. Chilipirea, C., Petre, A.-C., Groza, L.-M., Dobre, C., & Pop, F. (2017). An integrated architecture for future studies in data processing for smart cities. *Microprocessors and Microsystems, 52*, 335–342.
81. Enayet, A., Razzaque, M. A., Hassan, M. M., Alamri, A., & Fortino, G. (2018). A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities. *IEEE Communications Magazine, 56*, 110–117.
82. Plageras, A. P., Psannis, K. E., Stergiou, C., Wang, H., & Gupta, B. B. (2018). Efficient IoT-based sensor BIG data collection–processing and analysis in smart buildings. *Future Generation Computer Systems, 82*, 349–357.
83. Syafrudin, M., Alfian, G., Fitriyani, N. L., & Rhee, J. (2018). Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. *Sensors, 18*, 2946.
84. El-Hasnony, I. M., Mostafa, R. R., Elhoseny, M., & Barakat, S. I. (2021). Leveraging mist and fog for big data analytics in IoT environment. *Transactions on Emerging Telecommunications Technologies*. https://doi.org/10.1002/ett.4057
85. Jindal, A., Kumar, N., & Singh, M. (2020). A unified framework for big data acquisition, storage, and analytics for demand response management in smart cities. *Future Generation Computer Systems, 108*, 921–934.
86. Hussain, M. M., Beg, M. S., & Alam, M. S. (2020). Fog computing for big data analytics in IoT aided smart grid networks. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-020-07538-1

87. Zhou, Z., Yu, H., & Shi, H. (2020). Human activity recognition based on improved Bayesian convolution network to analyze health care data using wearable IoT device. *IEEE Access, 8*, 86411–86418.
88. Sengupta, S., & Bhunia, S. S. (2020). Secure data management in cloudlet assisted IoT enabled e-health framework in smart city. *IEEE Sensors Journal, 20*, 9581–9588.
89. Babar, M., & Arif, F. (2019). Real-time data processing scheme using big data analytics in internet of things based smart transportation environment. *Journal of Ambient Intelligence and Humanized Computing, 10*, 4167–4177.
90. Hong-Tan, L., Cui-hua, K., Muthu, B., & Sivaparthipan, C. (2021). Big data and ambient intelligence in IoT-based wireless student health monitoring system. *Aggression and Violent Behavior*. https://doi.org/10.1016/j.avb.2021.101601
91. Paul, A., Ahmad, A., Rathore, M. M., & Jabbar, S. (2016). Smartbuddy: Defining human behaviors using big data analytics in social internet of things. *IEEE Wireless communications, 23*, 68–74.
92. Gohar, M., Ahmed, S. H., Khan, M., Guizani, N., Ahmed, A., & Rahman, A. U. (2018). A big data analytics architecture for the internet of small things. *IEEE Communications Magazine, 56*, 128–133.
93. Armoogum, S., & Li, X. (2019). Big data analytics and deep learning in bioinformatics with hadoop. In *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Elsevier, pp. 17–36.
94. Demchenko, Y., Turkmen, F., de Laat, C., Hsu, C. H., Blanchet, C., & Loomis, C. (2017). Cloud computing infrastructure for data intensive applications. In *Big Data Analytics for Sensor-Network Collected Intelligence*. Elsevier, pp. 21–62.
95. Wu, X., Zheng, W., Xia, X., & Lo, D. (2021). Data quality matters: A case study on data label correctness for security bug report prediction. *IEEE Transactions on Software Engineering*. https://doi.org/10.1109/TSE.2021.3063727
96. Erraissi, A., & Belangour, A. (2018). Data sources and ingestion big data layers: Meta-modeling of key concepts and features. *International Journal of Engineering & Technology, 7*, 3607–3612.
97. Ji, C., Shao, Q., Sun, J., Liu, S., Pan, L., Wu, L., et al. (2016). Device data ingestion for industrial big data platforms with a case study. *Sensors, 16*, 279.
98. Isah, H., & Zulkernine F (2018) A scalable and robust framework for data stream ingestion. In *2018 IEEE International Conference on Big Data (Big Data)*. pp. 2900-2905
99. Dai, H.-N., Wong, R.C.-W., Wang, H., Zheng, Z., & Vasilakos, A. V. (2019). Big data analytics for large-scale wireless networks: Challenges and opportunities. *ACM Computing Surveys (CSUR), 52*, 1–36.
100. Chawla, H., & Khattar, P., (2020). Data ingestion. In *Data Lake Analytics on Microsoft Azure*. Springer, pp. 43–85.
101. Sankaranarayanan, S., Rodrigues, J. J., Sugumaran, V., & Kozlov, S. (2020). Data flow and distributed deep neural network based low latency IoT-edge computation model for big data environment. *Engineering Applications of Artificial Intelligence, 94*, 103785.
102. Davoudian, A., Chen, L., & Liu, M. (2018). A survey on NoSQL stores. *ACM Computing Surveys (CSUR), 51*, 1–43.
103. Cao, B., Sun, Z., Zhang, J., & Gu, Y. (2021). Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing. *IEEE Transactions on Intelligent Transportation Systems, 22*, 3832–3840.
104. Sonbol, K., Özkasap, Ö., Al-Oqily, I., & Aloqaily, M. (2020). EdgeKV: Decentralized, scalable, and consistent storage for the edge. *Journal of Parallel and Distributed Computing, 144*, 28–40.
105. Akanbi, A., & Masinde, M. (2020). A distributed stream processing middleware framework for real-time analysis of heterogeneous data on big data platform: case of environmental monitoring. *Sensors, 20*, 3166.
106. Harb, H., Mroue, H., Mansour, A., Nasser, A., & Motta Cruz, E. (2020). A hadoop-based platform for patient classification and disease diagnosis in healthcare applications. *Sensors, 20*, 1931.
107. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. *Future Generation Computer Systems, 91*, 620–633.
108. Alves, J. M., Honório, L. M., & Capretz, M. A. (2019). ML4IoT: A framework to orchestrate machine learning workflows on internet of things data. *IEEE Access, 7*, 152953–152967.
109. Oğur, N. B., Al-Hubaishi, M., & Çeken, C. (2022). IoT data analytics architecture for smart healthcare using RFID and WSN. *ETRI Journal, 44*, 135–146.
110. Bashir, M. R., Gill, A. Q., Beydoun, G., & Mccusker, B. (2020). Big data management and analytics metamodel for IoT-enabled smart buildings. *IEEE Access, 8*, 169740–169758.
111. Chhabra, G. S., Singh, V. P., & Singh, M. (2018). Cyber forensics framework for big data analytics in IoT environment using machine learning. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-018-6338-1
112. Vögler, M., Schleicher, J. M., Inzinger, C., & Dustdar, S. (2017). Ahab: A cloud-based distributed big data analytics framework for the internet of things. *Software: Practice and Experience, 47*, 443–454.
113. Farmanbar, M., & Rong, C. (2020). Triangulum city dashboard: An interactive data analytic platform for visualizing smart city performance. *Processes, 8*, 250.
114. Ghallab, H., Fahmy, H., & Nasr, M. (2020). Detection outliers on internet of things using big data technology. *Egyptian Informatics Journal, 21*, 131–138.
115. Lan, K., Fong, S., Song, W., Vasilakos, A. V., & Millham, R. C. (2017). Self-adaptive pre-processing methodology for big data stream mining in internet of things environmental sensor monitoring. *Symmetry, 9*, 244.
116. He, X., Wang, K., Huang, H., & Liu, B. (2018). QoE-driven big data architecture for smart city. *IEEE Communications Magazine, 56*, 88–93.
117. Singh, A., Garg, S., Batra, S., Kumar, N., & Rodrigues, J. J. (2018). Bloom filter based optimization scheme for massive data handling in IoT environment. *Future Generation Computer Systems, 82*, 440–449.
118. Yu, W., Liu, Y., Dillon, T., Rahayu, W., & Mostafa, F. (2021). An integrated framework for health state monitoring in a smart factory employing IoT and big data techniques. *IEEE Internet of Things Journal, 9*, 2443–2454.
119. Zhang, Q., Zhu, C., Yang, L. T., Chen, Z., Zhao, L., & Li, P. (2017). An incremental CFS algorithm for clustering large data in industrial Internet of Things. *IEEE Transactions on Industrial Informatics, 13*, 1193–1201.
120. Shaji, B., Lal Raja Singh, R., & Nisha, K. (2022). A novel deep neural network based marine predator model for effective classification of big data from social internet of things. *Concurrency and Computation: Practice and Experience*. https://doi.org/10.1002/cpe.7244
121. Al-Osta, M., Bali, A., & Gherbi, A. (2019). Event driven and semantic based approach for data processing on IoT gateway devices. *Journal of Ambient Intelligence and Humanized Computing, 10*, 4663–4678.
122. Deng, X., Jiang, P., Peng, X., & Mi, C. (2018). An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in Internet of Things. *IEEE Transactions on Industrial Electronics, 66*, 4672–4683.

123. Yao, X., Wang, J., Shen, M., Kong, H., & Ning, H. (2019). An improved clustering algorithm and its application in IoT data analysis. *Computer Networks, 159*, 63–72.
124. Mansour, R. F., Abdel-Khalek, S., Hilali-Jaghdam, I., Nebhen, J., Cho, W., & Joshi, G. P. (2021). An intelligent outlier detection with machine learning empowered big data analytics for mobile edge computing. *Cluster Computing*. https://doi.org/10.1007/s10586-021-03472-4
125. Karyotis, V., Tsitseklis, K., Sotiropoulos, K., & Papavassiliou, S. (2018). Big data clustering via community detection and hyperbolic network embedding in IoT applications. *Sensors, 18*, 1205.
126. Chui, K. T., Liu, R. W., Lytras, M. D., & Zhao, M. (2019). Big data and IoT solution for patient behaviour monitoring. *Behaviour & Information Technology, 38*, 940–949.
127. Song, C.-W., Jung, H., & Chung, K. (2019). Development of a medical big-data mining process using topic modeling. *Cluster Computing, 22*, 1949–1958.
128. Khan, M., Iqbal, J., Talha, M., Arshad, M., Diyan, M., & Han, K. (2018). Big data processing using internet of software defined things in smart cities. *International Journal of Parallel Programming*. https://doi.org/10.1007/s10766-018-0573-y
129. Gohar, M., Muzammal, M., & Rahman, A. U. (2018). SMART TSS: Defining transportation system behavior using big data analytics in smart cities. *Sustainable cities and society, 41*, 114–119.
130. Anbarasan, M., Muthu, B., Sivaparthipan, C., Sundarasekar, R., Kadry, S., Krishnamoorthy, S., et al. (2020). Detection of flood disaster system based on IoT, big data and convolutional deep neural network. *Computer Communications, 150*, 150–157.
131. Luo, X., Oyedele, L. O., Ajayi, A. O., Monyei, C. G., Akinade, O. O., & Akanbi, L. A. (2019). Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands. *Advanced Engineering Informatics, 41*, 100926.
132. Hossain, M. A., Ferdousi, R., Hossain, S. A., Alhamid, M. F., & El Saddik, A. (2020). A novel framework for recommending data mining algorithm in dynamic iot environment. *IEEE Access, 8*, 157333–157345.
133. Safa, M., & Pandian, A. (2021). Intelligent big data analytics model for efficient cardiac disease prediction with IoT devices in WSN using fuzzy rules. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-021-08788-3
134. Alsaig, A., Alagar, V., Chammaa, Z., & Shiri, N. (2019). Characterization and efficient management of big data in IoT-driven smart city development. *Sensors, 19*, 2430.
135. Tang, R., & Fong, S. (2018). Clustering big IoT data by metaheuristic optimized mini-batch and parallel partition-based DGC in Hadoop. *Future Generation Computer Systems, 86*, 1395–1412.
136. Kotenko, I., Saenko, I., & Branitskiy, A. (2018). Framework for mobile internet of things security monitoring based on big data processing and machine learning. *IEEE Access*. https://doi.org/10.1109/ACCESS.2018.2881998
137. Wang, T., Bhuiyan, M. Z. A., Wang, G., Rahman, M. A., Wu, J., & Cao, J. (2018). Big data reduction for a smart city's critical infrastructural health monitoring. *IEEE Communications Magazine, 56*, 128–133.
138. Kaur, I., Lydia, E. L., Nassa, V. K., Shrestha, B., Nebhen, J., Malebary, S., et al. (2021). Generative adversarial networks with quantum optimization model for mobile edge computing in IoT big data. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-021-08706-7
139. Lakshmanaprabu, S., Shankar, K., Ilayaraja, M., Nasir, A. W., Vijayakumar, V., & Chilamkurti, N. (2019). Random forest for big data classification in the internet of things using optimal features. *International journal of machine learning and cybernetics, 10*, 2609–2618.
140. Ullah, F., Habib, M. A., Farhan, M., Khalid, S., Durrani, M. Y., & Jabbar, S. (2017). Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare. *Sustainable Cities and Society, 34*, 90–96.
141. Manogaran, G., Varatharajan, R., Lopez, D., Kumar, P. M., Sundarasekar, R., & Thota, C. (2018). A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting system. *Future Generation Computer Systems, 82*, 375–387.
142. Hendawi, A., Gupta, J., Liu, J., Teredesai, A., Ramakrishnan, N., Shah, M., et al. (2019). Benchmarking large-scale data management for internet of things. *The Journal of Supercomputing, 75*, 8207–8230.
143. Mo, Y. (2019). A data security storage method for IoT under hadoop cloud computing platform. *International Journal of Wireless Information Networks, 26*, 152–157.
144. Tu, L., Liu, S., Wang, Y., Zhang, C., Li, P. (2019). An optimized cluster storage method for real-time big data in internet of things. *The Journal of Supercomputing*. 1–17.
145. Tripathi, A. K., Sharma, K., Bala, M., Kumar, A., Menon, V. G., & Bashir, A. K. (2020). A parallel military-dog-based algorithm for clustering big data in cognitive industrial internet of things. *IEEE Transactions on Industrial Informatics, 17*, 2134–2142.
146. Alelaiwi, A. (2017). A collaborative resource management for big IoT data processing in Cloud. *Cluster Computing, 20*, 1791–1799.
147. Meerja, K. A., Naidu, P. V., & Kalva, S. R. K. (2019). Price versus performance of big data analysis for cloud based internet of things networks. *Mobile Networks and Applications, 24*, 1078–1094.
148. Wang, T., Liang, Y., Zhang, Y., Arif, M., Wang, J., & Jin, Q. (2020). An intelligent dynamic offloading from cloud to edge for smart IoT systems with big data. *IEEE Transactions on Network Science and Engineering*. https://doi.org/10.1109/TNSE.2020.2988052
149. Vasconcelos, D., Andrade, R., Severino, V., & Souza, J. D. (2019). Cloud, fog, or mist in IoT? That is the question. *ACM Transactions on Internet Technology (TOIT), 19*, 1–20.
150. Jamil, B., Ijaz, H., Shojafar, M., Munir, K., & Buyya, R. (2022). Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*. https://doi.org/10.1145/3513002
151. Javadzadeh, G., & Rahmani, A. M. (2020). Fog computing applications in smart cities: A systematic survey. *Wireless Networks, 26*, 1433–1457.
152. Cao, B., Zhang, J., Liu, X., Sun, Z., Cao, W., Nowak, R. M., et al. (2021). Edge–cloud resource scheduling in space–air–ground-integrated networks for internet of vehicles. *IEEE Internet of Things Journal, 9*, 5765–5772.
153. Linaje, M., Berrocal, J., & Galan-Benitez, A. (2019). Mist and edge storage: Fair storage distribution in sensor networks. *IEEE Access, 7*, 123860–123876.
154. Mehdipour, F., Noori, H., & Javadi, B. (2016). Energy-efficient big data analytics in datacenters. In *Advances in Computers*. Vol. 100. Elsevier, pp. 59–101.
155. Zhou, L., Mao, H., Zhao, T., Wang, V. L., Wang, X., & Zuo, P. (2022). How B2B platform improves Buyers' performance: Insights into platform's substitution effect. *Journal of Business Research, 143*, 72–80.
156. García-Magariño, I., Lacuesta, R., & Lloret, J. (2017). Agent-based simulation of smart beds with Internet-of-Things for exploring big data analytics. *IEEE Access, 6*, 366–379.

157. Bi, Z., Jin, Y., Maropoulos, P., Zhang, W.-J., & Wang, L. (2021). Internet of things (IoT) and big data analytics (BDA) for digital manufacturing (DM). *International Journal of Production Research*. https://doi.org/10.1080/00207543.2021.1953181
158. Ahmed, I., Ahmad, M., Jeon, G., & Piccialli, F. (2021). A framework for pandemic prediction using big data analytics. *Big Data Research, 25*, 100190.
159. Puschmann, D., Barnaghi, P., & Tafazolli, R. (2016). Adaptive clustering for dynamic IoT data streams. *IEEE Internet of Things Journal, 4*, 64–74.
160. Bu, F. (2018). An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT. *Future Generation Computer Systems, 88*, 675–682.
161. Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT. *Information Fusion, 39*, 72–80.
162. Lavalle, A., Teruel, M. A., Maté, A., & Trujillo, J. (2020). Improving sustainability of smart cities through visualization techniques for big data from IoT devices. *Sustainability, 12*, 5595.
163. Li, P., Chen, Z., Yang, L. T., Zhang, Q., & Deen, M. J. (2017). Deep convolutional computation model for feature learning on big data in internet of things. *IEEE Transactions on Industrial Informatics, 14*, 790–798.
164. Patterson, E. K., Gurbuz, S., Tufekci, Z., & Gowdy, J. N. (2002). CUAVE: A new audio-visual database for multimodal human-computer interface research. In *2002 IEEE International conference on acoustics, speech, and signal processing*, pp. II-2017-II-2020.
165. Zhang, Q., Yang, L. T., & Chen, Z. (2015). Deep computation model for unsupervised feature learning on big data. *IEEE Transactions on Services Computing, 9*, 161–171.
166. Cauteruccio, F., Cinelli, L., Corradini, E., Terracina, G., Ursino, D., Virgili, L., et al. (2021). A framework for anomaly detection and classification in Multiple IoT scenarios. *Future Generation Computer Systems, 114*, 322–335.
167. Liang, W., Li, W., & Feng, L. (2021). Information security monitoring and management method based on big data in the internet of things environment. *IEEE Access, 9*, 39798–39812.
168. Vahdat, S. (2022). A review of pathophysiological mechanism, diagnosis, and treatment of thrombosis risk associated with COVID-19 infection. *IJC Heart & Vasculature*. https://doi.org/10.1016/j.ijcha.2022.101068
169. Abbasi, S., Naderi, Z., Amra, B., Atapour, A., Dadkhahi, S. A., Eslami, M. J., et al. (2021). Hemoperfusion in patients with severe COVID-19 respiratory failure, lifesaving or not? *Journal of Research in Medical Sciences, 26*, 34.
170. Li, W., Chai, Y., Khan, F., Jan, S. R. U., Verma, S., Menon, V. G., et al. (2021). A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system. *Mobile Networks and Applications, 26*, 234–252.
171. Biswas, R. (2022). Outlining big data analytics in health sector with special reference to Covid-19. *Wireless Personal Communications, 124*, 2097–2108.
172. Wu, X., Zhang, Y., Wang, A., Shi, M., Wang, H., & Liu, L. (2020). MNSSp3: Medical big data privacy protection platform based on Internet of things. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-04873-z
173. Elhoseny, M., Abdelaziz, A., Salama, A. S., Riad, A. M., Muhammad, K., & Sangaiah, A. K. (2018). A hybrid model of internet of things and cloud computing to manage big data in health services applications. *Future generation computer systems, 86*, 1383–1394.
174. Jan, M. A., He, X., Song, H., & Babar, M. (2021). Machine learning and big data analytics for IoT-enabled smart cities. *Mobile Networks and Applications, 26*, 156–158.
175. Liu, Z., Wang, Y., & Feng, J. (2022). Vehicle-type strategies for manufacturer's car sharing. *Kybernetes*. https://doi.org/10.1108/K-11-2021-1095
176. Khan, M. A., Siddiqui, M. S., Rahmani, M. K. I., & Husain, S. (2021). Investigation of big data analytics for sustainable smart city development: An emerging country. *IEEE Access, 10*, 16028–16036.
177. Sivaparthipan, C., Muthu, B. A., Manogaran, G., Maram, B., Sundarasekar, R., Krishnamoorthy, S., et al. (2020). Innovative and efficient method of robotics for helping the Parkinson's disease patient using IoT in big data analytics. *Transactions on Emerging Telecommunications Technologies, 31*, e3838.
178. Yang, L., Xiong, Z., Liu, G., Hu, Y., Zhang, X., & Qiu, M. (2021). An analytical model of page dissemination for efficient big data transmission of C-ITS. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2021.3134557
179. Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A review of machine learning and IoT in smart transportation. *Future Internet, 11*, 94.
180. Guo, J., Liu, R., Cheng, D., Shanthini, A., & Vadivel, T. (2022). Urbanization based on IoT using big data analytics the impact of internet of things and big data in urbanization. *Arabian Journal for Science and Engineering*. https://doi.org/10.1007/s13369-021-06124-2
181. Shao, N. (2022). Research on architectural planning and landscape design of smart city based on computational intelligence. *Computational Intelligence and Neuroscience*. 2022.
182. Jia, T., Cai, C., Li, X., Luo, X., Zhang, Y., & Yu, X. (2022). Dynamical community detection and spatiotemporal analysis in multilayer spatial interaction networks using trajectory data. *International Journal of Geographical Information Science*. https://doi.org/10.1080/13658816.2022.2055037
183. Kahveci, S., Alkan, B., Mus'ab, H. A., Ahmad, B., & Harrison, R. (2022). An end-to-end big data analytics platform for IoT-enabled smart factories: A case study of battery module assembly system for electric vehicles. *Journal of Manufacturing Systems, 63*, 214–223.
184. Nitti, M., Girau, R., & Atzori, L. (2013). Trustworthiness management in the social internet of things. *IEEE Transactions on knowledge and data engineering, 26*, 1253–1266.
185. Shahab, S., Agarwal, P., Mufti, T., & Obaid, A. J. (2022). SIoT (social internet of things): A review. *ICT Analysis and Applications*. https://doi.org/10.1007/978-981-16-5655-2_28
186. Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks, 56*, 3594–3608.
187. Baldassarre, G., Giudice, P. L., Musarella, L., & Ursino, D. (2019). The MIoT paradigm: Main features and an "ad-hoc" crawler. *Future Generation Computer Systems, 92*, 29–42.
188. Meghana, J., Hanumanthappa, J., & Prakash, S. S. (2021). Performance comparison of machine learning algorithms for data aggregation in social internet of things. *Global Transitions Proceedings, 2*, 212–219.
189. Lo Giudice, P., Nocera, A., Ursino, D., & Virgili, L. (2019). Building topic-driven virtual iots in a multiple iots scenario. *Sensors, 19*, 2956.
190. McCall, J. (1994). Quality factors, encyclopedia of software engineering. (vol. 2, p. 760). New York: Wiley
191. Boehm, B., & In, H. (1996). Identifying quality-requirement conflicts. *IEEE software, 13*, 25–35.
192. Grady, R. B. (1992). *Practical software metrics for project management and process improvement*: Prentice-Hall, Inc.
193. Talia, D. (2019). A view of programming scalable data analysis: From clouds to exascale. *Journal of Cloud Computing, 8*, 1–16.

194. Firmani, D., Mecella, M., Scannapieco, M., & Batini, C. (2016). On the meaningfulness of "big data quality." *Data Science and Engineering, 1*, 6–20.
195. Jabbar, S., Ullah, F., Khalid, S., Khan, M., & Han, K. (2017). Semantic interoperability in heterogeneous IoT infrastructure for healthcare. *Wireless Communications and Mobile Computing, 2017*
196. Rialti, R., Marzi, G., Caputo, A., & Mayah, K. A. (2020) Achieving strategic flexibility in the era of big data. *Management Decision*.
197. Roy, D., Srivastava, R., Jat, M., & Karaca, M. S. (2022). A complete overview of analytics techniques: descriptive, predictive, and prescriptive. *Decision intelligence analytics and the implementation of strategic business management,* 15–30.
198. Rahul, K., Banyal, R. K., Goswami, P., & Kumar, V. (2021). Machine learning algorithms for big data analytics. In *Computational Methods and Data Engineering*, Springer, pp. 359–367.
199. Nti, I. K., Quarcoo, J. A., Aning, J., & Fosu, G. K. (2022). A mini-review of machine learning in big data analytics: Applications, challenges, and prospects. *Big Data Mining and Analytics, 5*, 81–97.
200. Rajendran, R., Sharma, P., Saran, N. K., Ray, S., Alanya-Beltran, J., & Tongkachok, K. (2022) An exploratory analysis of machine learning adaptability in big data analytics environments: A data aggregation in the age of big data and the internet of things. In *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pp. 32–36.
201. Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., et al. (2019). Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. *Sensors, 20*, 109.
202. Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing, 237*, 350–361.
203. Prastyo, D. D., Khoiri, H. A., Purnami, S. W., Fam, S.-F., & Suhermi, N. (2020). Survival support vector machines: A simulation study and its health-related application. *Supervised and Unsupervised Learning for Data Science* (pp. 85–100). Cham: Springer.
204. Pink, C. M. (2016). Forensic ancestry assessment using cranial nonmetric traits traditionally applied to biological distance studies. In *Biological Distance Analysis*, Elsevier, pp. 213–230.
205. Lu, W. (2019). Improved K-means clustering algorithm for big data mining under Hadoop parallel framework. *Journal of Grid Computing*. https://doi.org/10.1007/s10723-019-09503-0
206. Zheng, W., Liu, X., & Yin, L. (2021). Research on image classification method based on improved multi-scale relational network. *PeerJ Computer Science, 7*, e613.
207. Goswami, S., & Kumar, A. (2022). Survey of deep-learning techniques in big-data analytics. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-022-09793-w
208. Roni, M., Karim, H., Rana, M., Pota, H., Hasan, M., & Hussain, M. (2022). Recent trends in bio-inspired meta-heuristic optimization techniques in control applications for electrical systems: A review. *International Journal of Dynamics and Control*. https://doi.org/10.1007/s40435-021-00892-3
209. Swayamsiddha, S. (2020). Bio-inspired algorithms: principles, implementation, and applications to wireless communication. In *Nature-Inspired Computation and Swarm Intelligence*. Elsevier, pp. 49–63.
210. Ni, J., Wu, L., Fan, X., & Yang, S. X. (2016). Bioinspired intelligent algorithm and its applications for mobile robot control: a survey. *Computational intelligence and neuroscience, 2016*.
211. Game, P. S., & Vaze, D. (2020). Bio-inspired Optimization: metaheuristic algorithms for optimization. *arXiv preprint* arXiv:2003.11637.
212. Romero, C. D. G., Barriga, J. K. D., & Molano, J. I. R. (2016) Big data meaning in the architecture of IoT for smart cities. In *International Conference on Data Mining and Big Data*, pp. 457–465.
213. Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. S. (2017). Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys (Csur), 50*, 1–37.
214. Granat, J., Batalla, J. M., Mavromoustakis, C. X., & Mastorakis, G. (2019). Big data analytics for event detection in the IoT-multicriteria approach. *IEEE Internet of Things Journal, 7*, 4418–4430.
215. Xiong, Z., Zhang, Y., Luong, N. C., Niyato, D., Wang, P., & Guizani, N. (2020). The best of both worlds: A general architecture for data management in blockchain-enabled Internet-of-Things. *IEEE Network, 34*, 166–173.
216. Oktian, Y. E., Lee, S.-G., & Lee, B.-G. (2020). Blockchain-based continued integrity service for IoT big data management: A comprehensive design. *Electronics, 9*, 1434.
217. Liu, F., Zhang, G., & Lu, J. (2020). Multisource heterogeneous unsupervised domain adaptation via fuzzy relation neural networks. *IEEE Transactions on Fuzzy Systems, 29*, 3308–3322.
218. Dong, J., Cong, Y., Sun, G., Fang, Z., & Ding, Z. (2021). Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2021.3128560
219. Zenggang, X., Xiang, L., Xueming, Z., Sanyuan, Z., Fang, X., Xiaochao, Z., et al. (2022). A service pricing-based two-stage incentive algorithm for socially aware networks. *Journal of Signal Processing Systems*. https://doi.org/10.1007/s11265-022-01768-1
220. Benhamaid, S., Lakhlef, H., & Bouabdallah, A. (2021) Towards energy efficient mobile data collection in cluster-based IoT networks. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 340-343.
221. Sun, W., Lv, X., & Qiu, M. (2020). Distributed estimation for stochastic Hamiltonian systems with fading wireless channels. *IEEE Transactions on Cybernetics*.
222. Lv, Z., Qiao, L., & You, I. (2020). 6G-enabled network in box for internet of connected vehicles. *IEEE transactions on intelligent transportation systems, 22*, 5275–5282.
223. Xifilidis, T., & Psannis, K. E. (2022). Correlation-based wireless sensor networks performance: The compressed sensing paradigm. *Cluster Computing, 25*, 965–981.
224. Mohammadi, A., Ciuonzo, D., Khazaee, A., & Rossi, P. S. (2022). Generalized locally most powerful tests for distributed sparse signal detection. *IEEE Transactions on Signal and Information Processing over Networks, 8*, 528–542.
225. Aziz, A., Osamy, W., Khedr, A. M., El-Sawy, A. A., & Singh, K. (2020). Grey Wolf based compressive sensing scheme for data gathering in IoT based heterogeneous WSNs. *Wireless Networks, 26*, 3395–3418.
226. Djelouat, H., Amira, A., & Bensaali, F. (2018). Compressive sensing-based IoT applications: A review. *Journal of Sensor and Actuator Networks, 7*, 45.
227. Wang, K., Zhang, B., Alenezi, F., & Li, S. (2022). Communication-efficient surrogate quantile regression for non-randomly distributed system. *Information Sciences, 588*, 425–441.
228. Lee, G. H., Han, J., & Choi, J. K. (2021). MPdist-based missing data imputation for supporting big data analyses in IoT-based

applications. *Future Generation Computer Systems, 125*, 421–432.

229. Zhang, F., Zhai, J., Shen, X., Mutlu, O., & Du, X. (2021). POCLib: A high-performance framework for enabling near orthogonal processing on compression. *IEEE Transactions on Parallel and Distributed Systems, 33*, 459–475.
230. Abualigah, L., Diabat, A., & Elaziz, M. A. (2021). Intelligent workflow scheduling for big data applications in IoT cloud computing environments. *Cluster Computing, 24*, 2957–2976.
231. Naas, M. I., Lemarchand, L., Raipin, P., & Boukhobza, J. (2021). IoT data replication and consistency management in fog computing. *Journal of Grid Computing, 19*, 1–25.
232. Ma, Z., Zheng, W., Chen, X., & Yin, L. (2021). Joint embedding VQA model based on dynamic word vector. *PeerJ Computer Science, 7*, e353.
233. Rahouma, K. H., Aly, R. H., & Hamed, H. F. (2020). Challenges and solutions of using the social internet of things in healthcare and medical solutions—a survey. *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications* (pp. 13–30). Cham: Springer.
234. Corradini, E., Nicolazzo, S., Nocera, A., Ursino, D., & Virgili, L. (2022). A two-tier Blockchain framework to increase protection and autonomy of smart objects in the IoT. *Computer Communications, 181*, 338–356.
235. Pincheira, M., Antonini, M., & Vecchio, M. (2022). Integrating the IoT and blockchain technology for the next generation of mining inspection systems. *Sensors, 22*, 899.
236. Tchagna Kouanou, A., Tchito Tchapga, C., Sone Ekonde, M., Monthe, V., Mezatio, B. A., Manga, J., et al. (2022). Securing data in an internet of things network using blockchain technology: smart home case. *SN Computer Science, 3*, 1–10.
237. Ursino, D., & Virgili, L. (2020). An approach to evaluate trust and reputation of things in a Multi-IoTs scenario. *Computing, 102*, 2257–2298.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Arezou Naghib received her B.Sc. in computer engineering, software engineering, from Islamic Azad University (IAU), Khoy branch, Iran in 2009; the M.Sc. in Computer Science, from IAU, Shabestar Branch, Iran in 2013. From 2018, she is a Ph.D. student in computer engineering at Science and Research Branch, IAU (SRBIAU), Tehran. Arezou is currently a faculty member in College of Skills and Entrepreneurship, Urmia Branch, IAU, Iran. Her research interests include data mining, big data analytics, IoT, and programming.

Nima Jafari Navimipour received his B.Sc., M.Sc., and Ph.D. degrees in computer engineering from Islamic Azad University, Iran, in 2008, 2009, and 2014, respectively. Dr. Navimipour also is a technical committee member, guest editor, and associate editor of some high-ranked journals such as IET Quantum Communication, Optik, Journal of Management & Organization, Computer Communication, Cluster Computing, and Kybernetes. Furthermore, he is a chair member of many prestigious conferences and a reviewer of several high-ranked journals. He has been giving invited tutorials/talks in IEEE conferences and has been invited to give lectures in different universities. Nima also won the Publons Top Peer Review Awards in 2018 and 2019. Dr. Navimipour has been featured among the World's Top 2% Scientists List, according to a conducted study by US-based Stanford University in 2020. Nima is also a senior member of IEEE, IEEE Communications Society and IEEE Young Professionals. His research interests include cloud and distributed computing, Internet of Things (IoT), Software-Defined Networking (SDN), information systems, computational intelligence, evolutionary computing, and quantum computing. He has published many papers in various journals and conference proceedings as well as supervising/co-supervising several Ph.D. and Master's students in these research areas.

Mehdi Hosseinzadeh received his B.Sc. degree in computer hardware engineering from IAU, Dezful Branch, Iran in 2003. He also received his M.Sc. and the Ph.D. degree in computer system architecture from the SRBIAU, Tehran, Iran in 2005 and 2008, respectively. Mehdi is currently an associate professor in Pattern Recognition and Machine Learning Lab, Gachon University, Korea. He is the author/co-author of more than 120 publications in technical journals and conferences, and his research interests include IoT, SDN, information technology, data mining, big data analytics, E-Commerce, E-Marketing, and social networks.

Arash Sharifi received the B.Sc. degree in computer hardware engineering from IAU South Tehran Branch, M.S degree and Ph.D. degree in artificial intelligence from SRBIAU, in 2007 and 2012 respectively. He is currently head of computer engineering department of SRBIAU. His current research interests include image processing, machine learning and deep learning.