ARTICLE IN PRESS

Digital Communications and Networks xxx (xxxx) xxx

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

IoT data analytic algorithms on edge-cloud infrastructure: A review

Abel E. Edje [a,b,*], M.S. Abd Latiff [a], Weng Howe Chan [a,c]

[a] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia
[b] Department of Computer Science, Delta State University, Abraka, P.M.B 01 Abraka, Delta State, Nigeria
[c] UTM Big Data Centre, Ibnu Sina Institute of Scientific and Industrial Research, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia

ARTICLE INFO

Keywords:
Internet of things
Cloud platform
Edge
Analytic algorithms
Processes
Network communication protocols

ABSTRACT

The adoption of Internet of Things (IoT) sensing devices is growing rapidly due to their ability to provide real-time services. However, it is constrained by limited data storage and processing power. It offloads its massive data stream to edge devices and the cloud for adequate storage and processing. This further leads to the challenges of data outliers, data redundancies, and cloud resource load balancing that would affect the execution and outcome of data streams. This paper presents a review of existing analytics algorithms deployed on IoT-enabled edge cloud infrastructure that resolved the challenges of data outliers, data redundancies, and cloud resource load balancing. The review highlights the problems solved, the results, the weaknesses of the existing algorithms, and the physical and virtual cloud storage servers for resource load balancing. In addition, it discusses the adoption of network protocols that govern the interaction between the three-layer architecture of IoT sensing devices enabled edge cloud and its prevailing challenges. A total of 72 algorithms covering the categories of classification, regression, clustering, deep learning, and optimization have been reviewed. The classification approach has been widely adopted to solve the problem of redundant data, while clustering and optimization approaches are more used for outlier detection and cloud resource allocation.

1. Introduction

The Internet of Things (IoT)-enabled edge cloud is an emerging ubiquitous network infrastructure that provides various distributed services in every aspect of human life. Smart devices such as sensors, microcontrollers, mobile phones, local servers, and the cloud can interact with each other to perform tasks and share information. As the popularity and extensive use of IoT-enabled edge cloud increases over the years, more sensor data will be generated, and various IoT-enabled edge cloud applications will be implemented to provide quality services to end-users regardless of their geographical location.

IoT sensor devices are typically used to capture events that are sents to other connected devices and systems over Internet and other communication networks. IoT sensors are characterized by the generation of dynamic, heterogeneous, inaccurate, and weakly semantic data streams over time. However, the massive data streams cannot be processed due to limited storage and computational resources. Therefore, the generated data streams are offloaded to the edge device(s) or cloud data center for further processing and analysis. The cloud data center provides massive storage and processing power to handle large amount of data. However, it is challenged by issues such as latency distance and bandwidth, which are highly required to process real-time data streams retrieved from IoT sensing devices. Consequently, the edge device(s) are designed to address these challenges, which are made up of clusters of interconnected physical servers located in close proximity to the IoT sensor devices.

Cloud resource allocation for the processing of IoT sensory data streams tends to improve the efficiency and data quality through the use of various algorithms (e.g., supervised and unsupervised machine learning, optimization, and deep learning). This has further simulated the rapid adoption of data-driven analytics and cloud resource allocation algorithms to solve problems of data outliers, redundancies, and resource load balancing in IoT-enabled edge cloud infrastructure [1]. An anomaly or outlier is a data instance that is significantly different from the rest of the instances, as if it was retrieved from a different source. On the other hand, redundancy refers to duplicate or repeated sensed data or events captured over time. Such data is not considered useful and can negatively impact an application' performance and consume massive resources (such as storage, memory, and compute). Load balancing ensures that the workload of IoT application requests (e.g., data analysis or

* Corresponding author. Department of Computer Science, Delta State University, Abraka, P.M.B 01 Abraka, Delta State, Nigeria.
E-mail addresses: eaedje2@graduate.utm.my (A.E. Edje), shaffie@utm.my (M.S. Abd Latiff), cwenghowe@utm.my (W.H. Chan).

https://doi.org/10.1016/j.dcan.2023.10.002
Received 29 December 2019; Received in revised form 10 September 2023; Accepted 6 October 2023
Available online 13 October 2023

Please cite this article as: Abel E. Edje et al., *Digital Communications and Networks*, https://doi.org/10.1016/j.dcan.2023.10.002

filtering processes) is evenly distributed across available cloud resources to achieve efficient execution, with minimum resource utilization, and completion time.

Conversely, to the best of our knowledge, most of the existing literature 'reviews in this field, are yet to fully explore the use of data-driven analytic-enabled cloud resource allocation algorithms for the execution of sensory data streams on IoT-enabled edge cloud Infrastructure. Therefore, it is crucial to investigate the existing data-driven analytic-enabled cloud resource allocation algorithms that are used to address the challenges of data outliers, data redundancy, and cloud resource load balancing on IoT-enabled edge cloud infrastructure. The contribution of this work is as follows.

1) A detailed analysis of the algorithms to resolve outlier and redundancy issues in sensory data, highlighting the strengths and weaknesses of each algorithm in tabular form.
2) A detailed analysis of cloud resource allocation algorithms to address resource load balancing challenges, for optimal execution of outlier and redundant sensory data.
3) The identification and discussion of the various algorithms to perform their respective functions. Also, to compare their level of usage in the IoT-edge cloud infrastructure.
4) We also highlight and discuss the various network communication protocols that govern the interaction between the three-tiered IoT-enabled edged cloud architecture described in previous research.
5) Detailed current and potential challenges that pave the way for future research directions in this field are discussed in this paper.

The remainder of this paper is structured as follows. Section 2 introduces the background information about IoT-enabled edge cloud computing and the characteristics of IoT sensing data. Section 3 discusses the research methodology used to update the current research survey understudy. Section 4 discusses the existing literature surveys in this field. Section 5 presents the analysis of existing algorithms deployed for resolving data outliers, data redundancy, and load balancing-related issues in IoT-enabled edge cloud infrastructure. Section 6 discusses the processes adopted by existing algorithms and network communication protocols that govern the interaction between the three-layer architecture of the IoT-enabled edge cloud infrastructure. Section 7 presents the current challenges that pave the way for future research directions. Finally, Section 8 presents a general discussion based on result of the research survey and ends with concluding remarks.

2. Background

IoT technology has come a long way in recent years. The concept of IoT was introduced by Kevin Ashton in 1999 and has been widely Auto-ID Center. IoT is a worldwide network of interconnected devices addressable with standard communication protocols, with the Internet as the convergence point [2,3]. Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSNs) are the most widely used IoT sensing devices since their inception. RFID is composed tag and reader used to identify and track an object anywhere and anytime. It is used in the courier and logistic transportation industry to track goods in transit. The WSNs consist of multiple sensor nodes deployed for environmental monitoring. WSNs communicate cooperatively and forward aggregated data to the network sink node or control system for further processing [4]. Both sensing devices can be integrated for better sensing and tracking of objects by collecting information such as object locations, movement, and temperature.

Over the past decade, IoT sensor devices have experienced tremendous advances in development. Currently, IoT sensor devices pre-process, store, and transmit sensed data directly to the internet without any human intervention. Unlike WSN, IoT sensor devices do not communicate with each other or inter-networked to transmit their sensed data to a connected sink node. This emerging IoT sensing device is called a smart sensor, which is with electronics that can perform multiple logic functions, two-way communication, make decisions, store sensed information for future analysis, or offload it directly to the Internet [5]. Therefore, the limitations of WSNs which include input offset and span variation, cross sensitivity, and nonlinearity are automatically corrected by the smart sensor processor. IoT sensing devices generate massive data that is dynamic and heterogeneous. In addition, the rapid rate at which unstructured and semi-structured data is being generated is a common problem. There are four main characteristics of IoT sensed data namely multi-source high heterogeneity, sensing data inaccuracy, and weak semantic data with low-level and enormous data dynamicity. Sensing data inaccuracy refers to the information collected from IoT sensing devices, due to several limitations such as unreliable reading, which leads to data outliers. This brings about the complexity of using the sensed data directly for its purpose. Therefore, appropriate multi-dimensional and data processing techniques need to be adopted for accurate retrieval of sensed data.

Enormous data dynamics arise from interconnected multi-sensors, embedded in a large-scale environment. Communications between the various sensors always results in a large volume of data generated in real time, resulting in duplicate (redundant) data. Weak semantic data with a low level is attributed to the sensed data obtained from IoT sensing devices. This is due to the spatial-temporal correlation relationships of the sensed data. Therefore, the extraction of useful information from the massive data generated is needs to be performed in an event-driven perspective. The acquisition of sensed data from distributed sensor nodes varies from character to integer, video and audio streaming.

The provision of computational resources to store process sensed data, filtering or analysis cannot be handled by IoT sensor devices. This is due to the characteristics of sensed data, and the limited storage and computation resources of IoT sensor devices. However, the cloud platform has been used in recent years to address these limitations. Its large pool of data storage resources and high computation power on complex tasks leverages the limitations of the IoT sensor devices. The idea of cloud computing was initiated in 1951 when John Macarthy envisioned the importance of time-shared computers, to share hardware and software resources among multiple end-users with real time multi-tasking and programming.

Madhavaiah and Irfan [6], defined cloud computing as a technology-based business model, delivered as a service over the Internet, where software and hardware computing services are accessed virtually by end-users, based on-demand in a self-service perspective irrespective of their geographical location. There are three services offered by cloud platforms namely, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS is a web-based interface that allows end-users to access to cloud software applications; PaaS enables developers to have access to various development tools for the implementation of software applications on its platform. On the other hand, IaaS provides storage and computation processing power. These services can be accessed from Cloud Service Providers (CSPs) such as Google, IBM Salesforce, Amazon, and Microsoft.

Currently, the Google cloud platform provides intelligent IoT services that enable end-users to connect their physical IoT sensing devices to the platform and process, analyze and store the sensed data. The platform consists of fully managed cloud services and scalability, an integrated software stack for on-premises computing, and machine learning approaches for all IoT needs. Additionally, IBM launched its IBM IoT Connection Service in 2016 to formalize the use of IBM IoT for connected offerings on the cloud, which ingests and transforms sensory data obtained from sensors into meaningful insights. It also integrates the existing functionalities of IoT for electronic solutions available on its IBM Bluemix (an open standard for developing, managing, and running multiple applications) cloud platform with additional data storage, security, and monitoring functions. Microsoft introduces IoT services on its Cloud Azure platform namely, Azure IoT Hub and Azure IoT central.

IoT Hub is an open cloud platform that enables end-users to securely connect, monitor, and manage numerous devices to implement IoT applications. Azure IoT Central is an IoT SaaS solution that makes it explicit for end-users to connect, monitor, and manage the physical features of the IoT sensing devices. Even though the cloud may offer virtually unlimited resource storage and computational processing power to leverage the limitations of IoT devices, the long-distance network communication between them is a problem that needs a solution. In other words, the long-distance communication between both technologies due to bandwidth availability may hinder the prospect of their integration if not curtailed.

The Long-distance communication between them leads to latency and delay which can hinder timely responses in critical situations. For example, healthcare workers needs to constantly monitor patients in critical condition by equipping them with IoT sensing devices in their respective homes through the Internet provided by the cloud application layer. Another challenge is in the area of privacy and security. Owners of IoT sensing devices tend not to send their data to the cloud data storage center because of the unknown storage location. Recently, edge/gateway computing has been introduced lately to address these challenges. This distance is also responsible for the long delays that sometimes exits between the clients' IoT sensing devices and the traditional cloud [7,8].

Edge computing consists of clusters of servers that located close to the IoT sensing devices for timely response to service requests while conserving bandwidth consumption rate and latency delay. On the other hand, IoT sensing devices can offload their sensed data to the edge servers when the load exceed their capabilities. The proximity between edge and the IoT devices, provides an opportunity to control the latency delay between the IoT devices and the traditional cloud. In addition, the sensed data collected from IoT devices is stored and immediately processed by the edge servers, with only a fraction of the data being sent to a cloud data center for long-term processing. This results in reduced network load by conserving bandwidth transmission rate. Table 1 shows the characteristics of the IoT sensing devices, edge, and the conventional cloud data center, while

Fig. 1 shows the three-layer physical architecture of the investigated IoT-edge cloud infrastructure.

Table 1
Comparative features of IoT sensing device(s), edge and cloud platform.

S/N	Features	IoT sensing devices	Edge computing	Cloud computing
1	Components	Physical devices	Clusters of servers	Virtual resources
2	Storage capability	Minimum	Limited	Massive
3	Data availability	Source	Process	Process
4	Utilization of Network communication bandwidth rate	High, due to continuous event sensing	Minimum, due to the fact that sensed data is processed locally and stored in edge servers close to the IoT sensing devices	High bandwidth consumption due to the long distance between the cloud and IoT devices
5	Computational resource power	Limited	Limited	Unlimited
6	Deployment	Distributed	Decentralized	Centralized
7	Quality of service delivery in terms of timeliness	Continuous sensing	Faster, due to location proximity to IoT devices	Slower, due to the long distances between IoTs and cloud data-centers
8	Level of safety in data transmission operations	Minimal risk of data attack while in transits.	Minimal risk on data attack while in transits.	Long-distance communication between IoT and cloud pre-empts attacks on data while in transits
9	Resource and service location proximity for task execution	Not applicable	Edge servers usually close to IoT sensing devices	Remote datacenters are usually far away from IoT sensing devices

3. Research methodology

The research survey understudy was conducted with the support of the methodology utilized by Kitchenham and Charters [9]. The exploration of the literature contributions, covering the years 2011–2019 was obtained from the academic research databases, which were considered the most relevant to achieve the objectives of the current study. These databases include IEEE Xplore, Google Scholar, Springer, Scopus, and ScienceDirect. The search phrase ("internet of things data" OR "mining algorithm" OR "edge" and "storage resource provisioning" OR "IoT data" OR "cloud data center") was used to retrieve articles relevant to the current study. However, the results of the search query returned numerous research articles that were not relevant to the study.

The relevant articles not retrieved after the initial search were expected to be present in the referenced list of these results and were included in the analysis iteration. Research articles published only in English and contained in journals and conference proceedings were considered. The initial result yielded a total of 502 retrieved articles. Each article undergoes a series of quality assessment phases until it was finally selected. These phases are composed of four sequences which are highlighted as follows;

- Evaluate the title and exclude it if it does not conform to algorithms used in the IoT-enabled edge-cloud platform (current study).
- Read the abstract and discard it if it is not relevant to the current study
- Read and evaluate the introduction and conclusion, reject if the contribution is the same as other relevant articles.
- Analytically assess the research contribution quality and disqualify articles with low quality.

The considerations of articles accepted were considered based their degree of relevance to the current study. In addition, the writing quality, soundness, clarity, and credibility of the contributions made by each of the articles were considered.

A total of 85 articles scale through the quality assessment, which are highly relevant to the current research question. These 84 articles further subjected to the process of extraction to retrieve the desired information required to accomplish the objectives of the research study. The required information is highlighted below.

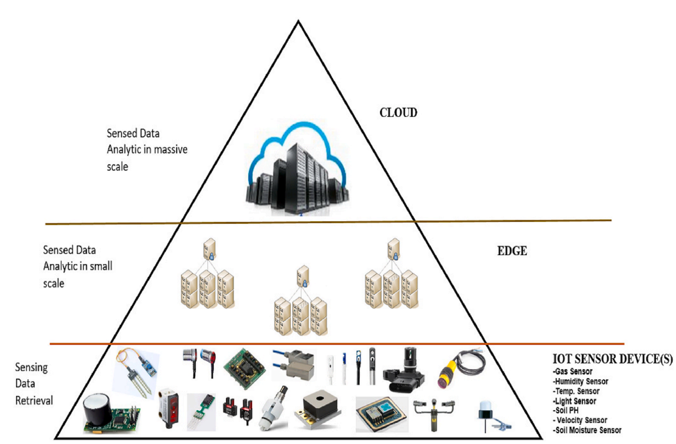

Fig. 1. IoT-enabled edge cloud architectural design.

- The algorithms used for outlier and redundant data detection.
- Allocation of resources to execute IoT application requests, problem resolution, and outcome.
- Performance evaluation processes adopted by each algorithm.
- Strengths and weaknesses of each algorithm.
- The network communication protocols govern the transmission of sensed data from IoT sensing devices to the edge and to the cloud storage server.
- The number of physical and virtual machines used to process application requests for IoT-sensed data (based on detection of outlier and redundant data) on the edge enabled cloud Infrastructure as a Service (IaaS) platform.

A total of 72 desired candidate articles emerged from the extraction process to be used in the current research under study. Fig. 2 presents a summary of the bibliometric data that includes 5 conferences and 67 Journal articles, for a total of 72 studies of the selected articles. It also shows that the number of studies increased over the years. Therefore, it shows the novelty and increasing interest in using algorithms for IoT data filtering/analysis based on the detection of outliers and redundant sensor data. In addition, the resource allocation algorithm is used to provide optimal computation and storage resources for the execution of sensory data filtering/analytic application requests on IoT-enabled edge cloud computing infrastructure. The remaining 13 articles are considered suitable for use as related research works in this field, which is discussed in the next section of this paper. Finally, 72 articles are qualitatively analyzed to synthesize the findings.

4. Related work

This section presents a brief description of previous literature review in this research field which motivated the current research study. The research survey conducted by Qiu et al. [10] is based on conventional and the latest machine learning algorithms for the processing and managing IoT big data. It discusses relevant machine learning algorithms in recent research such as the representation of learning, deep learning, distributed and parallel learning as well as active learning, and kernel learning. The challenges and possible solutions to machine learning algorithms for the processing of IoT data are also analyzed. Subsequently, the relationship between machine learning techniques and signal processing techniques used in the processing of IoT big data is highlighted and various open issues and research trends are outlined. Farahzadi et al. [11] studied the middleware technologies that are utilized in the Cloud of Things (CoT) platform. Firstly, the relevant features of middleware are discussed followed by the presentation of various architecture and service domains. It also explores the types of middleware that are appropriate for CoT-based platforms and outlines future challenges and issues in the design of CoT middleware. Cui et al. [12] present an overview of the application of machine learning techniques in the IoT domain. It discusses the current advances in applying machine learning techniques to IoT-related processes such as IoT device (s) identification, security, IoT edge computing infrastructure, traffic profiling, and network management. Also, research challenges and open issues of machine learning for IoT were extensively discussed.

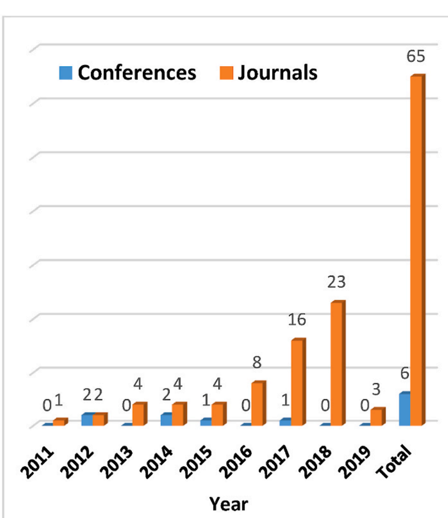

Fig. 2. Survey of previous researches on data analytics algorithms for IoT-based edge cloud.

An overview of several machine learning algorithms that tend to solve the challenges of IoT sensor data is presented in the research work of Mahdavinejad et al. [13]. It focused on the taxonomy of machine learning algorithms, describing how they have been used on IoT datasets to retrieve some relevant level of information. It also discussed the prospects and challenges of the algorithms for IoT data analysis, paving the way for the application of a Support Vector Machine (SVM) to Aarhus smart city traffic data as a use case for a more detailed investigation. Cai et al. [14] presented the recent achievements in the management, processing, and extraction of IoT big data by utilizing several existing algorithms. Thus, the algorithms are defined and described in terms of their significant features and capabilities, and the current challenges and opportunities associated with IoT big data are analyzed. Also, some typical examples and open issues in the application of algorithms for data acquisition are discussed. A thorough investigation of the use of mining algorithms in the management of IoT big data by Shadroo et al. [15]. It further identifies and discusses the architecture, framework, and applications of IoT big data. It also briefly discusses the algorithms used for the processing of IoT data in three categories which include descriptive, predictive, and classification.

In [16], a Novel Concentric Computing Model (CCM) is investigated for the use of IoT big data analytics applications. It discusses the sensing systems, and outer/inner gateway processors that make up CCM. In addition, it highlights current research work related to the IoT model for big data analytic techniques. It also describes the current challenges that need to be addressed for the deployment of CMM in the Internet of Things environments. Thus, various future research directions are presented such as dispatching of significant data, real-time fusion of streaming data, and data integration. Sharma and Wang [1] investigated the enablers for live data analytics in wireless IoT networks and storage provisioning by edge-enabled cloud computing environments. The framework for systematic processing between the cloud and the edge device(s) is discussed. It also highlights the networks and the available information in the cloud data center to support the edge computing units to meet various performance requirements of the wireless IoT networks. The key enablers in data analytics, such as NoSQL database and distributed file systems, to handle the unstructured IoT big data in the edge cloud are also discussed. In addition, machine learning techniques are used to extract relevant data. Related challenges and selected future research directions for researchers are also highlighted.

Recent advances in massive data analytics for IoT systems and the potential requirements for managing big data, as well as enabling analytic techniques enablers in IoT platforms [17]. Requirements such as IoT connectivity, storage capabilities, quality of service, and real-time services, and real-time analytics are discussed in detail. It explains the role of data analytics in IoT applications such as smart health, smart grid, and smart transportation as well as presents various open challenges as future research directions. Ge et al. [18] investigated big data technologies in several IoT domains to improve knowledge sharing across the IoT domains. It explained the similarities and differences between big data technologies and the analytics techniques (e.g., classification, filtering, compression, extraction, indexing, prediction, and storage) used in different IoT domains such as health, agriculture, and transportation to retrieve knowledge information. It further suggested how some big data technology deployed in a specific domain, can be re-used in different IoT domains. Also, a conceptual framework was formulated to identify the critical big data technologies across all the IoT domains reviewed.

The review of the task offloading scheme proposed for cloud computing, fog, and the IoT is discussed in Ref. [19]. It describes the middleware technologies (e.g., cloudlet, mobile edge, Micro datacenter, and Nano datacenter) that facilitate the offloading in edge-IoT infrastructure. It also presents research opportunities in offloading data streams in the fog and edge computing paradigm. Mohammadi et al. [20] presented a comprehensive overview of using a class of ground-breaking machine learning algorithms that can perform analytics and learning in the IoT domain. Detailed background information on various Deep Learning (DL) algorithms is presented, and specific research efforts that have used DL in the IoT domain are highlighted. The implementation approaches of DL on fog and cloud centers for the provisioning of IoT applications are also discussed. Fei et al. [21] studied several machine learning algorithms and how they are deployed on fog and cloud architectures for optimal processing and timely retrieval of data. In addition, the time complexity of the machine learning techniques used for IoT data stream analysis is highlighted. The challenges of deploying machine learning algorithms in the fog and cloud are also discussed, paving the way for future research directions.

Alam et al. [22] conducted a review on data fusion for IoT. It describes various mathematical techniques (e.g., probabilistic, artificial intelligence, and theory belief) use of IoT data analysis. It also detailed the prospects and challenges of each mathematical technique adopted in specific IoT environments (e.g. heterogeneous, distributed, object tracking, and nonlinear environments). In addition, future advances are discussed, including emerging area (autonomous vehicles, futuristic applications, and infotainment systems and smart cities) that would benefit immensely from data fusion and IoT.

The related research survey conducted by previous researchers summarized in Table 2, motivated the current research survey in this paper. Thus, we investigate the various algorithms used for IoT data filtering/analytics based on outlier detection, redundant sensed data elimination, and optimal load balancing of cloud resources allocated to execute the filtering/analytics-based IoT applications on edge-enabled cloud infrastructure. This is because the previous reviews have yet to provide substantial contributions to the aforementioned research problems in this research field. Also, the network communication protocols that govern the interaction and data transmission within the IoT, edge, and cloud layers are considered in this paper. These are discussed in detail in the following sections.

5. Analysis of algorithms on IoT-edge cloud

This section analyses the various analytic algorithms used for outlier and redundancy detection/elimination, as well as the allocation of resources to fulfill application requests in the cloud. Application symbolizes outlier detection, redundancy elimination, etc. We start with that of outliers, followed by redundancy and resources provisioning as follows;

5.1. Outlier detection algorithms

An outlier is a piece of data that does not conform to the rest of the data or follow the expected trend [23]. It is an essential feature of data mining, where the goal is to identify outliers or unusual data from a given data set. Outlier detection has been extensively studied in machine learning and statistics, and it is also known as anomaly detection, novelty detection, and deviation detection [24]. In particular, outlier detection in IoT-enabled edge cloud computing has been an aspect of great importance, as it becomes even more of interest due to the heterogeneity and dynamism of IoT sensor data. However, it has not been given the necessary attention and consideration, in the existing literature. The detection of outliers in sensory datasets is used for the removal of error data, the detection of faulty IoT sensing device(s), and detection

Table 2
Comparison of previous research surveys.

Author	Article Title	Contributions
Qiu et al. [10]	A survey of machine learning for big data processing	• Review of conventional and advanced machine learning methods for solving big data problems • Logical analysis of the challenges and potential solutions for leaning big data, based on the characteristics of big data • Open questions and research trends
Farahzadi et al. [11]	Middleware technologies for cloud of things: a survey	• Identification and explanation of IoT-Cloud middleware characteristics • Comparison of middleware architectures. • Middleware service domain e.g. information sharing and storage and communication • Comparison of sample middleware e.g. C-MOSDEN, ThingsWorx and Carriots • Challenges and issues in the Cloud of Things
Cui et al. [12]	A survey on application of machine learning for Internet of Things	• Description of possible supervised and unsupervised machine learning for traffic profiling • Identification of IoT devices (mobile phones and general IoT devices) using machine learning • Review on machine learning approaches for IoT system security based device and network security • Summary of IoT applications (e.g. health and industries) developed using machine learning • The use of machine learning approach for IoT network management and edge computing design • Challenges and open questions of the above reviewed areas
Mahdavinejad et al. [13]	Machine learning for Internet of things data analysis: a survey	• Describes the machine learning algorithms used to process data collected from IoT devices • Review of eight types of machine learning techniques used for IoT data analytics • Brief discussion of research trends and open questions
Cai et al. [14]	IoT-based big data storage systems in cloud computing: perspectives and challenges	• Analysis of cloud-based IoT application utility framework based on this data acquisition and processing • Discussion on the challenges of IoT data acquisition and methods used for data processing • Brief discussion on application module optimization based on architecture optimization, data storage optimization and data operation optimization
Shadroo et al. [15]	Systematic survey of big data and data mining in Internet of Things	• Review of tools used for IoT and big data processing

(continued on next page)

Table 2 (*continued*)

Author	Article Title	Contributions
		• Analysis of various techniques used for IoT device management/data mining • Brief discussion on open issues of IoT big data and mining methods
Rehman et al. [16]	Big data analytics in industrial IoT using a concentric computing model	• Review of the applicability of concentric computing model for big data analytics in IoT • Brief summary of communication and performance goals that can be achieved by adopting the concentric computing model in IoT • Highlighting some potential challenges and open issues that may lead to future research directions
Ahmed et al. [17]	The role of big data analytics in Internet of Things	• Review on the processing and key requirements of big data in IoT environment • Big data processing and analytics opportunities and the applicability of data analytics in IoT applications • Highlighting open research challenges of big data processing in IoT that leads to future research directions
Ge et al. [18]	Big data for Internet of Things: a survey	• Analyze the comparison of big data technologies in different IoT domains • Recommend the type of big data technology that can be used in other IoT domains • To shed more light on big data for each IoT domain • Framework to assist practitioner and researchers to adopt big data technologies that are commonly used in specific IoT domain
Aazam et al. [19]	Offloading in fog computing for IoT: review, enabling technologies, and research opportunities	Current technologies used for offloading in fog computing • Different requirements adopted by existing middleware technologies for offloading tasks in fog computing • Challenges that still need to be addressed for optimal performance of task offloading
Mohammadi et al. [20]	Deep learning for IoT big data and streaming analytics: a survey	• Leveraging deep learning in various IoT application domains Current methods for applying deep learning in a wide range of devices, from constrained to the fog and the cloud • Challenges and future research directions for the integration of deep learning and IoT applications
Fei et al. [21]	CPS data streams analytics based on machine learning for cloud and fog computing: a survey	• Machine learning methods used for IoT data processing in cyber-physical system applications • Time complexity of traditional machine learning strategies • Requirements for integrating machine learning methods into fog and cloud architecture
Alam et al. [22]	Data fusion and IoT for smart ubiquitous environments: a survey	• Mathematical techniques used for sensor data fusion • Review on special IoT environments such as heterogeneous and distributed environments • Challenges of individual mathematical techniques and IoT environments

of an event of interest [25]. These can only be achieved through the use of analytic algorithms which are discussed in detail, highlighting the processes employed by the algorithms to solve the prevailing problems, performances, and weaknesses of the algorithms, edge devices, and cloud IaaS resource(s) used to store and execute the algorithms, as indicated in Table 3. It also shows where the algorithms are deployed. For example, part of an algorithm may be implemented at the edge while the other part is located in the cloud. On the other hand, the entire part of the algorithm can be stationed either in the edge or cloud.

An adaptive Compressive Sensing-based (CS) autoregressive reconstruction algorithm is proposed for the sparsity of sensory data which varies in the temporal and spatial domain [26]. The autoregressive method is responsible for reconstructing false data retrieved from faulty sensor nodes. This is realized by exploiting the varying local spatial similarity in the sensed data set with an estimated parameter. If the sensed data exceeds the estimated parameter, it is classified as an anomaly or false data that needs to be reconstructed, otherwise the sensed data is classified as consistent data. The recovered data is then evaluated to determine whether additional measurements are needed to improve the reconstruction quality and whether the recovery process meets the expected accuracy. Furthermore, a combinational method is introduced to predict and identify the sparsity, which is incorporated into the CS to recover anomalous sensed data. Then, the recovered abnormal data are classified into two groups namely, error and external event by their identified patterns. The external event data is considered to reflect the actual activities in the environment and is preserved for further processing. On the other hand, the error data represent the physically sensed data which are discarded and replaced with their original normal readings.

However, the CS-based autoregressive reconstruction algorithm is not able to predict and identify an event that occurs a real-time event. It has been solved with the support of a model-based Multilayer Perceptron Classifier (MLPC) proposed by Stocker et al. [27]. The MLPC model is capable of obtaining knowledge that is represented in a semantic database by abstracting sensed data from the physical sensor layer on a real-time basis. At the initial stage, a band-pass filter is applied to pre-process the raw sensor data sample, after which the Multilayer Perceptron (MLP) neural network classifier is used to predict and classify various abstractions and events. The result of the classification process is transferred to the semantic database. However, the MLPC model is prone to a long non-automated learning process that requires domain experts to provide the model with sample data for the supervised learning process. This issue has been addressed in the research work of Ganz et al. [28], which introduces an approach that infers abstractions based on pattern representations. The approach is called the Sensor Symbolic Aggregation Approximation (SAX) algorithm, which is implemented to convert continuous sensor data into a compressed pattern representation. Firstly, the sensed data is normalized to have a standard deviation of 1 and a mean of 0, to facilitate the comparison of data points from different sources and to limit the volume of the sensed data sample. The sampled data is divided into two equal-sized windows by calculating the mean value of each window, resulting in the data sample being reduced

Table 3
Comparison of outlier detection techniques.

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
Data gathering in Sensor nodes through intelligent compressive sensing [26]	Adaptive compressive sensing-based autoregressive reconstruction algorithm	Classification	Abnormal sensing data	Improved accuracy and reduces Mean Squared Error and latency	Unable to predict and identify events that occur on a frequent basis	Remote Server	N/A	N/A	N/A
Making sense of sensor data using ontology: a discussion for residential building monitoring [27]	Knowledge- based multi neural network classifier	Classification	Predict and identify events that occur on a frequent basis	Accurate prediction of frequent anomaly sensing data on a real-time basis	Long non-automated learning process that relies on domain experts for the provisioning of sample data	Remote Server	N/A	N/A	N/A
Information abstraction for heterogeneous real world internet data [28]	Sensor symbolic aggregation approximation algorithm	Classification	Issue of minimizing the huge volume of sensing data	Improved accuracy, minimized data volume and latency	Abstraction accuracy still needs further improvement	Remote server	Yes	N/S	N/S
Smart outlier detection of wireless sensor network [29]	Fuzzy-based spatial-temporal approach	Classification	Detection of error and event outliers in local/global search space of the sampled data	Improved accuracy for error/event outliers with minimum false positive rate	Unable to self-check the prediction process using a mandatory perception data in an IoT platform	Remote Server	N/A	N/A	N/A
Non-parametric sequence-based Learning approach for outlier detection in IoT [30]	Non-parametric sequence learning algorithm	Classification	Problem of self-check identification using perception for error/event outliers detection	Enhanced classification accuracy with optimal detection of error/event outliers with less false positive rate	Difficult to detect outliers in global space data set increases is size	Remote Server	Yes	N/S	N/S
Cooperative sensor anomaly detection using global information [31]	Multivariate Gaussian-based principal component analysis	Classification	The inability to differentiate between erroneous and event data from inconsistent observations	Improved the Receiver Operating Characteristic (ROC) curves, true and false positive rates for detecting erroneously sensed data	Its static transformation is unable to realize optimal prediction of erroneous data on real-time basis	Remote Server	N/A	N/A	N/A
Recursive principal component analysis-Based data Outlier detection and sensor Data aggregation in IoT systems [32]	Recursive principal component analysis	Clustering	Inability to make optimal prediction of erroneous data in global space of massive sensed data sets	Improved aggregation with error growth and event detection accuracy	It is computationally intensive because it tends to adapt recursively to the changes in sensory data readings	Aduino Uno Microcontroller	Yes	N/S	N/S
A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases [33]	Logistic regression-based prediction algorithm	Regression	Ineffective classification of heart related disease symptoms	Enhanced classification accuracy rate based on specificity, and sensitivity	Inefficient aggregation of data	Mobile Phones Personal Server	Yes	N/S	N/S
Non-parametric sequence-based Learning approach for	Non-parametric sequence learning algorithm	Classification	Problem of self-check identification using	Enhanced classification accuracy with optimal	Difficult to detect outliers in global space data set increases is size	Remote Server	Yes	N/S	N/S

(continued on next page)

Table 3 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
outlier detection in IoT [30]			perception for error/event outliers detection	detection of error/event outliers with less false positive rate					
Cooperative sensor anomaly detection using global information [31]	Multivariate Gaussian-based principal component analysis	Classification	The inability to differentiate between erroneous and event data from inconsistent observations	Improved the Receiver Operating Characteristic (ROC) curves, true and false positive rates for detecting erroneously sensed data	Its static transformation is unable to realize optimal prediction of erroneous data on real-time basis	Remote Server	N/A	N/A	N/A
Recursive principal component analysis-Based data Outlier detection and sensor Data aggregation in IoT systems [32]	Recursive principal component analysis	Clustering	Inability to make optimal prediction of erroneous data in global space of massive sensed data sets	Improved aggregation with error growth and event detection accuracy	It is computationally intensive because it tends to adapt recursively to the changes in sensory data readings	Aduino Uno Microcontroller	Yes	N/S	N/S
A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases [33]	Logistic regression-based prediction algorithm	Regression	Ineffective classification of heart related disease symptoms	Enhanced classification accuracy rate based on specificity, and sensitivity	Inefficient aggregation of data	Mobile Phones Personal Server	Yes	N/S	N/S
A real IoT device deployment for e-health applications under lightweight communication protocols, activity classifier and edge data filtering [34]	Fuzzy-based human activity recognition classifier algorithm	Classification	The complexity of data overlapping in massive sensed data	Improved outliers/inliers detection accuracy with less computation resources	Unable to deal with missing data values	Mobile Phones	Yes	N/S	N/S
On the effect of adaptive and non-adaptive analysis of time-series sensory data [35]	Dynamic symbolic aggregation approximation (DSAX)	Clustering	The complexity of aggregating massive sensory data retrieved from various sources	Achieved optimal data aggregation quality for error data prediction	Unable to give insight knowledge about the sensing data retrieved regarding drifts and consistent data	Local Server	N/A	N/A	N/A
Adaptive clustering for dynamic IoT data streams [36]	Adaptive K-means clustering algorithm	Clustering	Deficiency in clustering streaming sensed data	Improved clustering accuracy based on Silhouette coefficient	Unable to consider the spatial dimension and correlation of the streaming data	Fog Server	N/A	N/A	N/A
Clustering of data streams with dynamic Gaussian mixture Models. an IoT application in industrial processes [37]	Gaussian-based dynamic probabilistic algorithm	Clustering	Inefficiency in clustering dynamic sensing data to detect drifts sensed data	Improved drift detection accuracy to the tune to 98.7% and sensitivity of 96% indicating that almost all detections are true positives	There are about as many Instances detected as turning points for concept drift	Local Server	N/A	N/A	N/A

(*continued on next page*)

Table 3 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
An automatic health monitoring system for patients suffering from voice complications in smart cities [38]	Linear prediction spectrum algorithm	Regression	Ineffective detection of voice disorder of patients	Efficient detection of voice disorders with sustained vowels and running speech based on improved accuracy	Not Specified	Local Server	N/A	N/A	N/A
Edge computing with Cloud for voice disorder assessment and treatment [39]	Convolution neural network algorithm	Deep learning	Inaccurate classification of voice disorder symptoms	Improved the classification accuracy in the detection of voice (event) disorder detection	High bandwidth consumption during sense data transmission from edge to the cloud	Cloudlet Servers	Yes	1	3
Fog assisted-IoT enabled patient health monitoring in smart homes [40]	Bayesian belief network algorithm	Deep learning	Delay in the classification of sensed data acquisition	Improved accuracy of classifying dataset with less time during classification process	Not considering the spatio-temporal correlations among sensed data set	Cloudlet Servers Wireless Routers	Yes	8	N/S
A new shelf life prediction method for farm products based on an agricultural IoT [41]	Back propagation learning algorithm	Deep learning	Issue of detecting and elimination of erroneous outliers in big sensed data	Effective filtering of normal sensed data from the abnormal ones	Not specified	Local Server	N/A	N/A	N/A
IoT big-data centered knowledge granule analytic and cluster framework for BI applications: a Case base analysis [42]	Enhanced knowledge granule clustering algorithm	Clustering	Challenges of clustering high complex knowledge granules for outlier detection	Improved the precision and accuracy of outlier detection	Unable to minimize the inter-cluster distances of sensed data	Remote Server	N/A	N/A	N/A
Fog intelligence for real-time IoT sensor data analytics [43]	Homoscedasticity measurement Leven's test feed-forward neural networks algorithm	Deep learning	Improper selection of threshold leading to partial classification	Enhanced classification accuracy, sensitivity and precision	Duplicate sensed data and high computationally intensive	Arduino Uno Microcontroller Local Server	N/A	N/A	N/A
Efficient and flexible algorithms for monitoring distance-based outliers over data streams [44]	Advance micro-cluster-based continuous outlier detection algorithm	Clustering	Inefficient outlier detection on frequent data stream and computation complexity	Improved outlier detection with minimum computational resource usage	Unable to addressed uncertainty of data streams, instances assigned existential probability	Local Server	Yes	N/S	N/S
Smartphone-based outlier detection: a complex event processing approach for driving behavior detection [45]	Complex event processing –based Z-score and box plot	Clustering	Computation resource complexity of constraints IoT devices	Improved accurate detection outliers from online data streaming with less usage of computation and memory resources	Weakness in identifying outliers for emergency scenarios due to lack of historical data	Fog Server	Yes	N/S	N/S
Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering [46]	Hyperellipsoidal clustering algorithm	Clustering	The Issue of high latency and energy consumption	Reduced energy consumption and latency while improving anomaly prediction accuracy	A need for further improvement on latency due to increase usage of computation resource	Fog Server	Yes	N/S	N/S

(continued on next page)

Table 3 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
Entropy outlier detection using semi-supervised approach with few positive examples [47]	Entropy Outlier Detection Semi-supervised (EODSP) algorithm	Clustering	Insufficient labeled data for training and limited positive labeled samples	Improved outlier detection accuracy compared to other existing approaches	Cannot be deployed in real time big sensing data due to its computational complexity	Laptop Pc (1.6 GHz and 1 GB RAM)	N/A	N/A	N/A
IPCA for network anomaly detection [49]	Iterative Principal Component Analysis (IPCA) algorithm	Clustering	Variability of feature scales and the issue of multiple number of dimension data set	Improved outlier detection efficiently mitigating the limitations of PCA	Computation complexity in iteratively updating distances of neighborhood	Remote Server	N/A	N/A	N/A
Real-time multiple event detection and classification using moving window PCA [50]	Moving Window Principal Component Analysis (MW-PCA) algorithm	Clustering	Issue of time variance in sensing data frequency	Improves the prediction and classification of outlier accuracy	Unable to disaggregate multiple loss of load and generation of events	Remote Server	Yes	N/S	N/S
Research on real time feature extraction method for complex manufacturing big data [51]	Robust Incremental Principle Component Analysis (RIPCA) algorithm	Clustering	Disaggregate multiple loss of load and generating of events	Improved outlier detection in real-time by reducing the dimension of big IoT dataset and usage of computation resource	Unable to determine the causes for the abnormal patterns	Remote Server	Yes	N/S	N/S

Footnote: N/A = Not Applicable, N/S=Not Specified.

to half its original size. As a result, the compressed data is reconstructed, allowing the adductive abstraction of the sensed data to discover events that occur over time. For example, the changes in temperature over a day from cold to warm to cold, which represents a frequent or stable temperature pattern. Therefore, newly observed states hidden from the pattern are classified as outliers.

Kamal [29], introduced a fuzzy algorithm that utilizes spatiotemporal similarity concept to detect outliers. However, could not provide the self-check identification using perception data which is highly required in an IoT Cloud-IaaS platform. It classifies the abnormal observation into error and event outliers. First, a data set generated by sensor nodes is computed on the first-order difference $|Si_2–Si_1|$. Then, the total difference is compared to the threshold value that is reached by the tolerance of the temperature sensor. Thus, if the total first-order difference does not exceed the threshold, the Si_2 data point is considered similar to other data points. Otherwise, an outlier is obtained when dissimilarity is observed on a data point. Second, the calculation is done based on the distance between neighboring sensor nodes to discover the spatial similarity between them. The Euclidean distance method used to compute the similarity or correlation measure between two points (x, y) that have identical transmission range and time proximity. Then, the spatial similarity threshold is obtained by computing the mean distance of all data points in the proximity time. If the Euclidean distance d(x, y) does not exceed the indicated threshold value, the data values at point X are identified as similar to that of the data values at point Y. Otherwise, an error outlier is detected as a faulty sensor reading.

Conversely, a Non-Parametric Sequence-based Learning (N-PSL) algorithm is proposed by Nesa et al. [30], predicting the outliers based on error event types. It considers the use of data perception for self-check detection both error and event outliers. The N-PSL algorithm is based on a gray relational analysis. In the initial stage, the sample data is normalized by calculating the average image of each sampled data. Then, the difference between each instance of the sequence image of the sensed data is computed. Also, the Influential Relative Grade (IRG) coefficients for each sequence (class) sensed data are calculated to retrieve the relative mass function in each respective class. Therefore, outliers are predicted as the classes with lower values while the inliers are classes with higher values. Furthermore, event outliers are detected by running the algorithm on the fused parameter (attribute) dataset, while the error type of outliers is obtained by running the algorithm on each parameter.

A Multivariate Gaussian-based Principal Component Analysis (MG-PCA) is designed in the research to predict erroneous sensing data among irregular observations, based on the characteristic pattern of different dimensional sequence data [31]. The MG is first applied to the retrieved sensed data set to determine the similarity among the data points. It identifies the time point when the error occurred and further retrieves the particular sensor node that is observed to be erroneous at a particular time. Consequently, the PCA utilizes the principal vectors to determine the differences between data patterns for detecting the sensor error readings that violate the inherent pattern extracted. However, the MG-PCA approach is limited by the inability to track variations in dynamic and heterogeneous sensing data due to its static transformation. This has been addressed by the Clustered-based Recursive Principal Component Analysis (CR-PCA) algorithm proposed in Ref. [32]. It initially aggregates the redundant sensed data while detecting the outliers. The spatially correlated sensed data retrieved from the cluster head sensor members are aggregated by extracting the principal components and identifying the possible data outliers with the support of an abnormal squared prediction error score, called the residual square. It recursively updates its parameters to adapt to the dynamics of the sensory data retrieved from the sensor devices.

A Logistic Regression-based prediction (LRP) algorithm is developed to detect patients with heart diseases by classifying clinical sensory data collected from IoT wearable devices [33]. Sensed data collected from wearable sensing devices are constantly monitored. If the data values exceed the reliable predicted value, it's considered abnormal, otherwise it becomes normal. Consequently, Santamaria et al. [34] proposed a fuzzy-based Human Activity Recognition (HAR) classifier algorithm to classify the sensed data into normal and abnormal activities of patients. The algorithm updates the classification process by initiating some constant values that are used to specify the number of clusters. It then selects a weighted component (fuzzier) and an initial membership matrix with some threshold values are selected. The weighted components regulate the class overlapping of the classes while assigning a data point to its cluster member. Furthermore, the threshold value is used to evaluate the convergence in the iterations of the classification process.

A Dynamic Symbolic Aggregation Approximation (SAX) is proposed for the adaptive and non-adaptive window size, in the segmentation of time sequence data stream with variation in real-time processing [35]. It divides the time sequence data set into equivalent segments and generates a string representation for each segment. First, the time sequence data is normalized to achieve a standard deviation and mean (average) of one, before being converted it to a Piecewise Aggregation approximation (PAA). Next, the data is divided into the desired number of windows and the average mean of the data falling in each window is calculated by the PAA so that the size can be reduced. Then, a discretization process is performed on the PAA coefficients (each window size) by mapping the PAA coefficients to breakpoints which are generated by the alphabet size (e.g. c), to determine the area of equal-size for retrieving the symbolic data representation. Puschmann et al. [36] developed an Adaptive K-means Clustering (AKC) for outlier detection. This is done by evaluating the dynamic sensor data and updating the cluster centroids according to the changes in the data stream at a given time. Clusters are formulated based on the similar features of the sensory data stream retrieved over time. New cluster(s) are formed based on changes in data features. For example, if an incoming streaming data has the feature types "Temp, Temp, Temp, Hum, and Hum ….n", obviously the Temp features will be allocated to the initial cluster. The appearance of the 'Hum" will trigger the creation of another new cluster which will contain the Hum feature data records.

Both the SAX and AKC approaches provide substantial assignment of sensory data instances to clusters but are unable to provide knowledge information (i.e. inconsistency or consistent manner) about the data and how it is assigned to each cluster. These problems have been addressed by a Gaussian-based Dynamic Probabilistic Clustering (GDPC) algorithm, proposed by Ref. [37]. It estimates the model parameters and any drifts in the data points. It further provides the membership likelihood of each data point to each cluster by utilizing the brier score. Brier score is used to determine the abnormality of subsequent probabilities from those objects or data points that are expected. Drifts or changes are detected when the parameter of sensed data value is above the predefined threshold value of the brier score. Such drifts are known as outliers. After drifts are detected, the brier score changes its behavior and stabilizes for incoming sensor data.

A Linear Prediction Spectrum algorithm is introduced in Ref. [38] for voice detection disorder, based on sensed data retrieved. It analyzes the energy variation in the spectrum to distinguish between disordered and normal voices. This is done by dividing the vocal track into various tubes from the glottis to the lips. It then performs an estimated analysis on the source signal using inverse filtering that triggers the computation of the spectrum. Furthermore, the estimated signal is utilized to determine the energy distribution in vowel and running speech for the detection of voice disorder. Muhammad et al. [39] develop a Deep Convolution Neural Network (CNN) algorithm for classifying the sensed data into two segments namely voice disorder and normal voice. It uses its input image consisting of blue, green, and red colors to classify the voice sampled data obtained from the IoT sensing devices. Therefore, the use of transfer learning and a fine-tuned approach is used to train the CNN for optimal detection of voice disorder and to speed up the classification process, due to the limited voice sampled data obtained from the IoT devices. Two output neurons were used to represent voice disorder detection, eight neurons were used for the voice disorder classification before being trained by fine-tuning the parameters for optimal detection of voice disorder from normal ones.

A Bayesian Belief Network (BBN) algorithm is proposed in Ref. [40] for the classification of sensory data. It classifies sensory data retrieved from patients into two classes namely, abnormal and normal. The retrieved data in the abnormal class indicates the severe or critical health status of the patients. On the other hand, the sampled data in the normal event class indicates the normality of patient's health status. A naïve Bayesian classification procedure known as conditional probability is used to achieve the classification process. Thus, a predefined value is set as the normal value, which indicates that the probability of all sampled data within the range of the predefined normal value will be classified as a normal class. Also, an abnormal class is obtained when the probability of having the sampled data value exceeds that of the normal event class. To improve the prediction process, an important set of attributes, namely the environment and the patient's history, were used. Thus, the abnormal class is transmitted to the cloud for further processing and analysis. Wu et al. [41], implemented a Back Propagation Learning (BP) algorithm for the classification of sensed data retrieved from sensing devices attached to agricultural crops. The sensed data are classified into abnormal and normal batches. The abnormal value or attributes are discarded while the normal values are further processed on the cloud platform. The normal values are further divided into low, normal, and high values based on predefined values consisting of -1 (low), 0 (normal), and 1 (high). The BP algorithm is then applied to accurately predict the crop yield. It multiplies the output and input data to obtain the gradient of the weight and places the weight in the opposite direction of the gradient by subtracting the ratio of it from the weight.

An Enhanced Knowledge Granule Clustering algorithm that is based on neuro-fuzzy analytic architecture is designed in Ref. [42]. It is used to extract complex knowledge granules from IoT sensory big data. First, the facts are arranged in an array based on the multiple rule system to obtain the knowledge granules for clustering. Each knowledge granule must be associated with a fitness tag, where the estimated value is present. This is done through the attributes of the knowledge granule where the initial mapping for a cluster is performed by the fitness value, followed by the next level mapping for sub-clusters under the previous cluster. In simple words, based on the fitness rule, two clusters are said to be similar if both have knowledge granules with homogenous attributes. The knowledge granules are mapped to individual clusters based on the attributes. Thus, the sub-cluster within a cluster is maintained for the fitness of the explicitly identified knowledge granules. For example, in cluster, let X1 be a knowledge granule such that X1 is mapped to sub-cluster ($G < 0.5$) if and only if $G(X1) < 0.5$; otherwise, X1 is mapped to sub-cluster ($G >= 0.5$). Thus, the G values of clusters and sub-cluster are strongly estimated by quantifying the outliers that are present. In addition, outliers that are present in the clusters and sub-clusters degrade the G values.

Furthermore, Raafat et al. [43] proposed a Homoscedasticity Measurement-based Leven's Test (HMLT)-based Feed-forward Neural Networks (FFNN) algorithm, for accurate classification of desired features of sensed data in the cloud. Sensed data retrieved from sensing devices is filtered. Then, the HTMLT is applied to extract dissimilarity features from the denoised signal, by observing the signal for sudden changes. Then, the extracted features are inputted into the FFNN to proceed with the classification process. The FFNN classifies the sensed data into abnormal and normal data. This is updated by sending the data from its input layer to the hidden layers. The neurons in the hidden layers are responsible for computing an activation function over the sum of input features, which are multiplied by a set of weight parameters. The, results are output as either normal or abnormal sensed data. Data is abnormal when there is a sudden change in the sensed data due to an external event.

An Advanced Micro-cluster-based Continuous Outlier Detection (AMCOD) algorithm is proposed in Ref. [44] for frequent monitoring of outliers in sensory data streams to improve efficiency and reduce storage resource utilization. An outlier 'A' is identified if the distance of 'B' instance (s) is greater than that of 'A'. Also, if the number of data instances or objects in the distant neighborhood of 'A' objects exceeds that of B, then 'A' is referred to an inlier. Efficiency is improved by using microclusters to minimize the number of distance calculations, memory size determination, and the number of data objects for the time window size. In addition, the arrival and departure of the data object is monitored to determine the degree of an outlier and safe inlier. At this stage, if the number of neighbors of a given data object of 'A' is higher than that of 'B', then 'A' becomes a safe inlier and not an outlier. Therefore, the use of computational and memory resources is reduced by discarding the outliers.

Conversely, the distance-based algorithm cannot run on devices that are challenged with low memory and computation resource. Vasconcelos et al. [45] solved the problem by introducing a Complex Event Processing Z-score and Box Plot approach to predict the outliers. The sensor data collected from the on-board vehicle and embedded mobile sensors are sent to the Complex Event Processing engine for preprocessing. It generates or extracts features (e.g. speed, acceleration, deceleration, mean deceleration, etc.) from the sensed data retrieved as evidence sensed data that best characterize chauffeur behavior. Then, patterns that significantly deviate from the evidence data are identified by the CEP rules. In addition, the Z-score method is used to assign a score to each piece of evidence by splitting the stream into sequence windows. Each window consists of a set of evidence sensed data. It then computes the standard mean deviation of the evidence in each window after which the Z-score distribution is assessed to classify the chauffeur's behavior. Moreover, the box plot method is deployed to avoid the computation complexity of pairwise distances for all evidence data by performing the computation for each evidence (dimension or feature) individually and correlating the outliers. It uses a threshold value to filter out all data instances that are inliers and those that are outliers.

A Hyperellipsoidal clustering algorithm is introduced by Ref. [46], to detect anomalies in the multimodal distribution of sensing data retrieved from end nodes. It accommodates heterogeneous sensing data ranging from linear to hyperspherical, with an automated mechanism to select the number of clusters. It also realizes a linear computation overhead regarding the number of data vectors processed. At the initial stage, a set of hyperellipsoidal clusters is obtained, by using the Ellipsoidal Neighborhood Outlier Factor (ENOF) to identify the ellipsoids that are drifting relative to their neighborhood to densities. Consequently, the ratio between the average neighborhood range density of neighbors and ellipsoids' neighborhood range determines the level of outlier score. Therefore, a threshold is calculated using the standard deviation of the ENOF scores and a parameter to determine the anomalous clusters. Thus, clusters with an ENOF score that is higher than the threshold are considered as anomalous clusters. The process of ENOF outlier detection is further illustrated in Fig. 3. Where the blue line represents the threshold value.

An Entropy Outlier Detection Semi-supervised (EODSP) algorithm is introduced in Ref. [47] for detecting the outliers in an unlabeled data set. Entropy is the degree of information and uncertainty of a random variable [48]. For instance, let y be a random variable, the entropy E(y) of the probability distribution g(y) on y = {y1 ….yn}; thus is given as a dataset of h instances with f-number of features, the entropy E(y) of a multivariable vector Yi is a random variable which is considered to be a member of y dataset. It consists of two strategies that are used to solve the problem of outlier prediction when there are limited positive data objects for training data. At the initial stage, the reliable negative data objects which are considered as inliers are extracted from positive samples and unlabeled data. Then, the distances between each point in the data set and positive objects are calculated. Therefore, the distance points that are higher than the threshold value from each data object are predicted as outliers. In addition, Delimargas et al. [49] proposed an Iterative Principal Component Analysis (IPCA) algorithm to detect data traffic anomalies in the network. The IPCA functions are as follows; the matrix of a data set M is obtained from the data trace, then the subtraction of the mean value of each feature is computed to formulate a new matrix of the data set, denoted NM. Then, each feature is divided by its standard deviation to obtain a normalized metric dataset Dn. The eigenvalues and eigenvectors of Dn are obtained by creating the correlation of its matrix (Dn). The eigenvector with the largest eigenvalue is considered the normal subspace while others are the anomalies. Therefore, it is updated iteratively when a new traffic data stream of packets is transmitted.

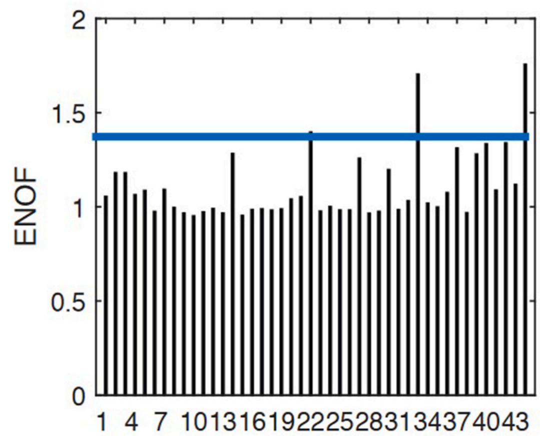

Fig. 3. An example of ENOF outlier detection.

Rafferty et al. [50] proposed a Moving Window Principal Component Analysis (MW-PCA) algorithm to obtain the threshold value for predicting an event that can adapt to the uncertainty behavior of a power system frequency for time variance. It learns on the initial window, containing a specific size of data frequency. Each newly normal data sample and that of PCA are calculated, updating their confidence limits to determine the subsequent new sample point. If the confidence limit of the initial normal data sample is less than or equal to the new sample and that of the initial data PCA confidence limit is also less than or equal to the new one then the system is considered to be operating normally and the moving window is updated to capture the new data sample. On the other hand, if both or either of the confidence bounds exceed the data point, it is automatically excluded, indicating the occurrence of (outliers). However, the aforementioned algorithms (EODSP, IPCA, and MW-PCA) can detect outliers from dynamic sensing data, but they are challenged with the inability of robustness to predict outliers in complex big and dynamic sensing data. It was solved using the support of a Robust Incremental Principle Component Analysis (RIPCA) algorithm, proposed by Kong et al. [51]. It uses the sliding window supported with an anti-K nearest neighbor method to compute the principal components of the sampled data set in the most current window to identify and discard outliers. The anti-K nearest neighbor is applied to the sliding window to update the current data and to predict the real-time data outliers. The anti-k nearest neighbor is a collection of data instances in a data set that considers an instance (a) as a K nearest neighbor. Therefore, the instances with at least of three anti-k-nearest neighbors are considered as the query outliers due to the anti-k neighbor in the current window.

5.2. Redundancy discovery

Data redundancy is the duplication or repetition of data, as shown in Fig. 4. It is a common problem in the IoT-enabled edge cloud domain. The sensed data generated by IoT sensing devices is massively dynamic, with redundancies due to the strong correlation between sensed data [52]. For example, certain data may appear multiple times in a dataset due to the repeated capture of an event by the sensor(s) within a certain

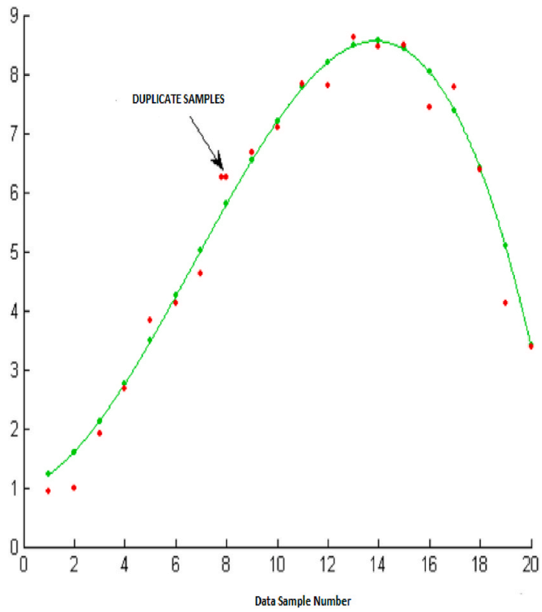

Fig. 4. Example of redundant data in dataset.

time period.

Most sensory data redundancies are considered irrelevant due to their negative impact on network and application performance. Just to mention a few, unnecessarily increases the size of the IoT device(s), limited storage, inconsistency, and data corruption. We present the filtering algorithms used to solve the problems related to sensor data redundancy on the IoT-enabled edge cloud platform, which are discussed as follows;

A Support Vector Machine Recursive Function Elimination-based Correlation Bias Reduction (SVMRFE+CBR) algorithm is developed in Ref. [53] to reduce the biased nature of SVM-RFE when a feature set consists of multiple similar features. Li et al. [54] initially implemented the SVM-RFE by using the requirement derived from the SVM coefficient to evaluate features and recursively discard features with limited requirements with the support of two different strategies namely Kernel and Wrapper. The Kernel strategy retains the dependencies among features, while the wrapper strategy does not use the cross-validation testing method on train samples as the requirement selection. It is also known to be efficient in terms of processing speed when dealing with different candidate features. It also makes maximum use of the training samples with minimum over-fitting.

The SVM-RFE is challenged to evaluate the feature(s) requirements, and their importance is underestimated due to excessive correlation between candidate features. However, the SVM-RFE is integrated with the Correlation Bias Reduction (CBR) strategy to improve the elimination of duplicate sensed processes. Therefore, the SVM-RFE+CBR solves the prevailing problems of SVM-RFE, by generating a representative feature with the highest demand of classified correlated features back into the existing feature class. First, the list of features to be eliminated during the first iteration is denoted as *Fout,* and the list of existing or relevant features is denoted as *Fin*. Two thresholds *Tc* and *Tg* are used to identify highly correlated feature classes in *Fout*. If there are more than *Tg* features whose coefficient with the highest demand is greater than *Tc*, they are identified as a group. Otherwise, if none of the group members are *Fin*, the features with the highest requirements are moved to *Fin*. Thus, this process is repeated for each feature in *Fout* until all the features have been removed. Szecowka et al. [55], proposed a Neural Network Sensitivity (NNS) approach for removing duplicate sensed data while maintaining the accuracy of the overall performance. An improved function was obtained with the support of the differential sequential coefficient of the neural network. However, NNS has some limitations consisting of a limited number of correlated features as inputs and the uncertain (confused) result based on overlapping (dependent) input features. These problems have been solved using the Fast Correlation-based Filter (FCBF) algorithm, implemented by Ref. [56]. It uses the Symmetrical Uncertainty (SU) to obtain the optimal and desired features among several features. The SU has threshold values ranging from 0 to 1, which used to evaluate the relationship between the feature class and the similarity between different features. Therefore, the variable can estimate the value of other variables if it's equal to 1. Otherwise, the two variables are independent if the value is equal to or less than the 0 mark. At the initial stage, it determines the association between feature and class subset with the support of C-Correlation while it performs the pairwise similarity among the features for the F-Correlation. Thus, feature redundancy is avoided during feature selection when the similarity between features and classes that satisfy the conditions of SU while searching for the relevant features, starting from the features with the highest SU values.

A Fractional-order Embedding Multi-set Canonical Correlations (FEMCCs) algorithm is introduced in Ref. [57] to resolve the eliminated data drifts from the consistently sampled data. In the initial stage, the covariance metrics are re-estimated using the fractional order to correct non-zero values and single values. Then, a fractional order is defined within-set and between-set scatter matrices to minimize the deviation or drift of the data sample matrices. It then extracts similar features from multiple sets of feature vectors obtained from the same objects. It then fuses the extracted similarity features together with the support of a fusion strategy, to form a discriminative feature vector for classification function. Haghighat et al. [58], proposed a Discriminant Correlation Analysis (DCA) approach to determine the class associations in the similarity data feature sets. It reduces the pairwise similarities of the corresponding feature sets simultaneously, discarding the feature similarities between classes and limiting the features belonging to different classes within each feature set. Then, the extracted features of interest from multiple classes are merged into a single class.

However, FEMCCs and DCA have some challenges. The minimized feature sets generated by FEMCCs seem to neglect certain correlation information among various feature sets which degrade its classification performance. On the other hand, DCA is deemed not to be effective as redundancies are still detected in the fused features because of the similarity requirement.

Both issues were resolved by utilizing Intra-class and Extra-class Discriminative Correlation Analysis (IEDCA-IRE) technique, proposed in Ref. [60]. It uses its Kernelize strategy to the intra-class similarity (pairswise correlation) and the similarity across various data features in the same class, to retain the relevant data in the fused data feature. After that, the irrelevant or duplicate data is eliminated. In simple words, it retains adequate dimensions of data features for class separation in each set of features and learned similarity features obtained by the discriminative structure. First, it generates a between-class scatter matrix via the nearest neighbor from both the extra-class and the intra-class. Then, the non-zero vectors of the corresponding nonzero values in the between-class matrix are identified. In addition, the maximization of feature correlation between-classes is maximized by computing the non-zero vectors with their corresponding values, which transforms the entire matrices. The Kernelized intra-class correlation is used to concatenate the transformed features into a fused feature vector as shown in Fig. 5(a and b), which leads to the elimination of irrelevant redundant features present in the fused feature vector.

However, FEMCCs and DCA do have some challenges. The minimized feature sets generated by FEMCCs seem to neglect certain correlation information among different feature sets, which degrades its classification performance. On the other hand, DCA is weak due to the discovery of redundant data in the fused features because of the similarity requirement.

Both problems were solved by using the Intra-class and Extra-class Discriminative Correlation Analysis (IEDCA-IRE) technique, proposed

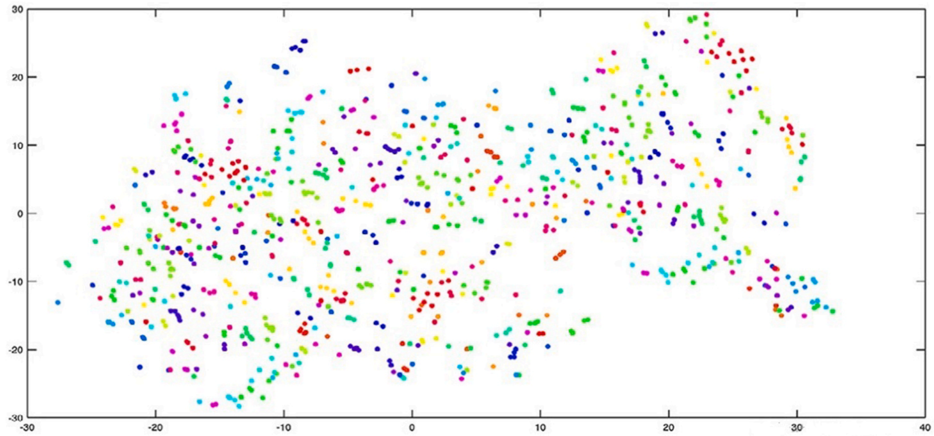

Fig. 5a. Original dataset.

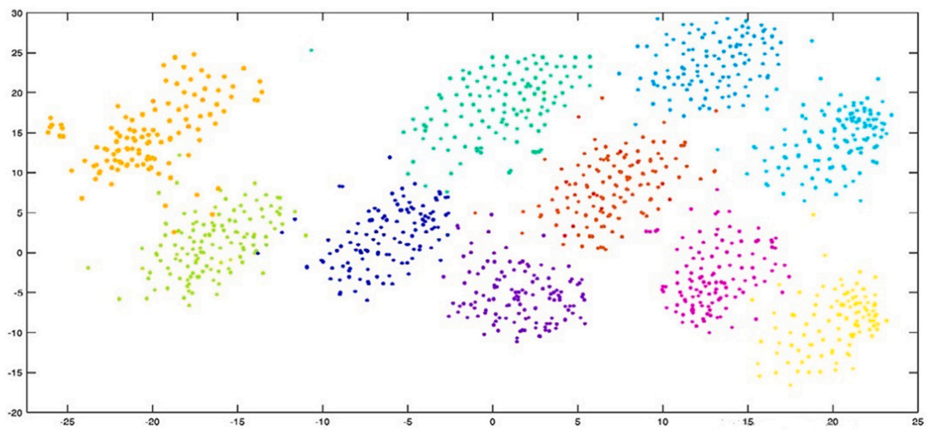

Fig. 5b. Fused features via intra-class/extra-class discriminative correlation.

in Ref. [60]. It uses its Kernelize strategy to search the intra-class similarity (pairswise correlation) and the similarity across different data features in the same class to retain the relevant data in the fused data feature. Then, the irrelevant or duplicate data is eliminated. In simple words, it retains adequate dimensions of data features for class separation in each set of features and learned similarity features obtained by the discriminative structure. Firstly, it generates a between-class scatter matrix via the neighbor in proximity to the extra-class and intra-class. Then, it identifies the non-zero vectors of the corresponding non-zero values in the between-class. Furthermore, the maximization of feature correlation between classes is obtained by computing the nonzero vectors with their corresponding values that transform the entire matrices. Therefore, the Kernelized intra-class correlation is used to concatenate the transformed features into a fused feature vector, as shown in Fig. 5(a and b). This results in the elimination of irrelevant redundant features present in the fused feature vector.

Jeffry-divergence (JD) and Inter-frame Correlation of Color Channels on Boolean Series–based Ensemble-based Support Vector Classification Algorithm is proposed in Ref. [59]. Thus, to minimize the massive amount of sensed data retrieved from camera sensing devices.

The obtained video frames (sensed data) are compared based on their color and structures. If similarities are detected between two or more frames, their divergence is computed using the color histogram to obtain the actual corresponding frame. Frames with high similarity measure are discarded. Then, a multi-fractal technique is used to discover the frames, based on different texture structures at different scales with local densities, to provide rich descriptors to categorize the structures of the frames. Then, an SVM is used to train each category of the frame the structure so that optimal informative frames (image) are extracted from the non-informative frames (image/data).

A Correlation Feature Selection-based Heuristic algorithm is introduced to address the problem of duplicate sensed data on edge-based cloud IaaS [61]. It uses the feature predictive performance and inter-correlation to guide its search for an optimal feature subset of sensed data. It also, considers the benefit of each feature of sensed data for predicting the class label, based on the level of inter-correlation among them. At the initial stage, it computes a matrix of feature-feature correlations and feature-class from the training data set. Then, an optimal search is performed to determine the feature subset space, by using the best first search technique to obtain the relevant features. Furthermore, Scale Invariant Feature Transform (SIFT) algorithm is developed by Yuan et al. [62], to manage the influx of sensing data retrieved from multimedia sensor nodes. The retrieved data are first fused by using the Laplace Pyramid Transform (LPT) method. Then, the different sizes of Gaussian Kernels (known to have more accurate scale transform) are selected to perform the scale transform of the fused data, to obtain the accurate candidate feature points. Therefore, the edge response points of low contrast and instability of the sensed data are discarded. Each feature point is allocated a direction by the gradient information of neighboring pixels to improve the accuracy of the feature point matching. Li et al. [63] propose a Center-symmetric Local Gabor Binary Pattern (CSLGBP) feature extraction algorithm to obtain the actual face image captured by camera sensor devices. The input face image is convolved with the Gabor kernels to retrieve the magnitude information of well-defined specific orientations and scales. The specified orientations at the same scale are accumulated to formulate a new scale feature. The features of each scale are computed using the CS-LBP descriptor from the retrieved Gabor scale images to extract and obtain the relevant image.

Linear Discriminate Analysis-based enhanced Support Vector algorithm is proposed in Ref. [64], to address the uncertainty with sensed image signal or data retrieved from camera sensor devices. It computes various characteristic features of the data sets and classifies the features present in the pre-processed sensed image signal. It also detects the Q wave, R wave, and S wave in the pre-processed input image signal to determine the various heartbeat levels (e.g., Left Bundle Branch Block, Right Bundle Branch Block, Premature Ventricular Contraction, and Premature Atrial Contractions) and classify them accordingly. The weighted kernel function computes the weight which is used to identify the R, Q, and S waves for optimal classification of the heartbeat levels. Consequently, the Incremental Fast Searching Clustering-based K-Mediods (ICFSKM) algorithm is introduced in Ref. [65], to discover the underlying patterns of the dynamic sensing data, by integrating the initial data patterns into the previous ones by using its combination operations. The cluster centers are continuously updated by the kmedoids upon the arrival of new sensing data. In simplicity, it maintains a set of clustered data with similar feature patterns, so it either creates new sets of clusters or assign them to the previous cluster upon new sensing data arrival.

A Blocks of Eigenvalues Algorithm for Time Series Segmentation (BEATS) is proposed to remove the duplicate sensed data from large datasets [66]. It divides the streams of time series data into 64 blocks, clustered the streams in square matrices and transforms them into frequency domain with the support of the Discrete Cosine Transform (DCT) technique. It is then quantized to obtain a finite data set. Then, the duplicate data is removed from the finite data set with the support of Eigen-values computation as shown in Fig. 6.

Consequently, Bu [67], develop an Efficient High-order Tensor Fuzzy C-means (EHOFCM) algorithm, based on the Canonical Polyadic Decomposition scheme for the clustering of IoT streaming data. The traditional fuzzy c-means (FCM) technique allocates each object or data record to two or more groups by computing a membership matrix. However, IoT-sensed big data is characterized by heterogeneous features, which is a notable drawback to the conventional FCM for the clustering of real-time IoT big data. The EHOFCM could solve the problems as follows. Each data point or object is convert from the vector

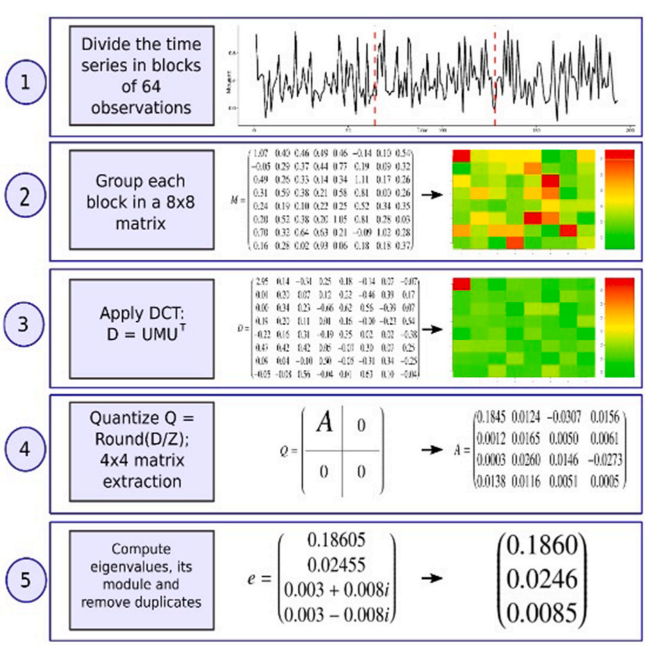

Fig. 6. Example of BEATS workflow [66].

space to its tensor format by a bijection function. Then they are aggregated into clustered groups based on their similarity features. In addition, the attributes of each object or data record are greatly reduced using the canonical polyadic decomposition scheme. Thus, to obtain an optimal compression rate as it reduces the huge volume of raw sensing data to some significant extent. Therefore, enabling the traditional fuzzy-c means to cluster the huge sensed data with low-end devices such as controllers and mobile phones.

Banag-Pseudo-cluster-based aggregation algorithm is developed in Ref. [68], to determine the exigency or criticality of various data collected from multiple sensor nodes. Data is aggregated into groups based on the level of their exigency at the edge (gateway) platform. Therefore, the data with the highest exigency value is aggregated first before the others. This is done repeatedly and systematically until all the sensed data are fused into their respective groups and sent to the cloud data center for further processing. Abawajy et al. [69] designed a Cobweb Expectation Maximization and K-means, which is also called the Rank Correlation Coefficient (RCC) algorithm for the clustering of ECG sensed data. First, it uses the fuzzy-based data fusion technique to aggregate only the relevant values of the sensed data and discard the others. Thus, the relevant data sets are grouped into different independent clusters. Then, a consensus function is used to combine the clusters to generate the final consensus cluster by partitioning all the elements or values of the dataset. Furthermore, Liu et al. [70] proposed a Two-step K-means Clustering (TKC) algorithm to cluster the image sensed data into two categories namely, Blurry and Clear Images. The Blurry images are discarded while the Clear Images are further processed at the edge platform. Clear image sensed data are segmented into two categories namely foreground (which contains the actual image data) and background (which contains useless image data) by utilizing the watershed segmentation function at the edge. This is done by using the Clear image and removing the background image, resulting in the updating of the foreground image.

Adaptive Moving Window Regression (AMWR) algorithm was developed by Akbar et al. [71], to determine the optimal training window size of streamed data, by using a Lomb-Scale time series analysis. For example, the temperature data retrieved over 24 h tend to contain repeated patterns or values. If the training window size of data used is equivalent to the optimal periodicity of the data, it will learn all the local patterns, resulting in more accurate prediction. In addition, the window sizes of data are predicted using the prediction horizon to ensure a certain level of prediction accuracy. This allows the window size prediction to be increased when the accuracy of the model is high and decreased when the performance of the prediction model decreases. Then, the output of the predicted block of data is transmitted to the Complex Event Processing engine in the form of an event tuple. Thus, applying predefined rules on the predicted block of data to detect or predict the complex event.

An Elephant Herd Optimization-based Linear Kernel Support Vector (EHO-LKSV) algorithm is proposed in Ref. [72], selecting the desired subset features from a dimensionally sensed data set. It greedily searches for the element space and determines a feasible feature subset to continuously improve the given input data, as it speeds up the computation time of the entire process. Furthermore, the retrieved feature subsets are classified into two different labels using a linear kernel support vector technique to train the different data sets for optimal prediction and accuracy results. Consequently, Wong et al. [73] proposed a novel Perceptually Important Points (PIP) algorithm, for the reduction of IoT time series sensing big data. It divides the sensed data into segments by identifying a set of important points either a set of local minima or local maxima out of the sensed data pools. At the initial stage, the time series feature alongside sensed data features is segmented into odd and even values, after which the similarity between features was determined by using the Jaccard similarity distance method. Similar instances with the same time retrieval value are eliminated across features, resulting in the reduction of the sensed data.

Hadoop Artificial Bee Colony (HABC) algorithm is developed in Ahmad et al. [74], for redundancy of sensed data. In the initial stage, the classified sensed data are placed into a subset according to their similarity characteristics by using the accuracy fitness values. In addition, the parameter of Medication Rate (MR) is used to extract features from neighboring subset data. Therefore, a random and uniform number (from 0 to 1) is generated for each data in each sensed data subset. If it is observed that the value is less than the MR, then the feature is inserted into a new subset. Otherwise, if the new subset happens to be better than the initial exploratory subset, it is considered as the last new subset. Thus, this process is repeated until the best feature subset is reached. A Deep Learning Long-short Term Memory (LSTM) algorithm is also proposed in Ref. [75], to predict the ground speed of aircraft landing, based on sensor data retrieved from the aircraft. It consists of six layers that are segmented into input, hidden, and output layers. A random forest algorithm is first used to classify the sensed data into twenty features. The input consists of one layer, the hidden consists of four layers and the output has only one layer. Consequently, the four hidden layers consist of 128, 64, 32, and 8 neurons while the output layer consists of one neuron, which is used to obtain the predictive value of ground speed.

Mohammadi et al. [76] proposed a Deep Reinforcement Learning (DRL) algorithm to aggregate sensed data with the same distance position, labeled and imputed in the same cluster. Sensed data are clustered based on their proximity level. It uses the variance auto-encoder function to identify the optimal data representing the closest distance information for locating the target object. Also, Yan et al. [77] proposed an Integrated Deep Auto-Encoder algorithm for the management of sensed data obtained from sensor devices. Data such as the state data recorded within a period at each sub-processes before the failure is retrieved from the DECG which is known as the historical information. The historical information is cleansed (e.g., filling missing data features) and divided into two categories, namely, distant records and recent records achieve an optimal prediction. The distant records symbolize the records that are far away from the current time moment, while the recent records indicate records that are close to the current time moment. Thus, the distant records are used to simulate the damaging trend, while the recent records are used to simulate the smoothing process of the recent change. Then, two outputs are fused and linear regression is performed to convert hidden or discrete records to predict the Remaining Useful Life (RUL) of production machines.

A deep learning based regression algorithm is proposed in Ref. [78]. It consists of eight layers which are further grouped into three sections namely lower layer, intermediate layers, and higher layers. The lower and intermediate layers are implemented in the edge servers while the higher layers are implemented in the cloud. The input sensed data (image of dog and cat) from the camera sensor devices are transferred to the lower layer in the edge servers for processing. The data are processed at the intermediate layer where a filter or feature detector is utilized to extract features to obtain the relevant data. This reduces the size of the input data to a significant size known as the relevant data. In addition, the reduced relevant data is transferred from the edge server to the cloud for further processing. The reduced data is passed to the higher layers (consisting of neurons) residing in the cloud server, where it is filtered (feature detector) to retrieve optimal data.

Hybrid Multilayer Perceptron Convolution Neural Network (MLP-CNN) algorithm is developed in Ref. [79], for the fusion and classification of sensed image data. Generally, it uses its fusion decision rule to fuse the output sensed data based on the CNN confidence value. The CNN confidence value is obtained by subtracting the maximum value of a vector from its mean value, resulting in the optimal membership classification. However, if the CNN confidence value is higher than an initial predefined threshold, it indicates that the CNN confidence is lower than another threshold. Thus, if the confidence of the CNN depends on the initial and the other threshold, then the fusion output selection with the higher confidence value is regarded as the actual classification result. Consequently, Liu et al. [80], develop a Convolutional Neural Network (CNN) algorithm to retrieve the desired sensed data on the cloud platform. It fine-tunes the sensed image dataset (image of various foods) to generate a fine-grained model that is used for the classification. Then, the fine-grained model is trained by Caffe. At the initial stage, the model is loaded into the memory, as the data (food image) is fed into the convolutional neural network as the input. Thus, the CNN features can be extracted by using the max-pooling and Rectified Linear-Unit (ReLU) layers, to reduce the data feature dimensions and speed up the convergence of the computing process.

Li et al. [81] proposed a Deep Convolutional Computation model (DCCM) algorithm to learn hierarchical features of sensed data by utilizing the tensor method, to extend the convolutional neural network from the vector space to the tensor space. Thus, the local features in the sensory data are optimally exploited and overfitting is avoided. Also, a tensor convolutional layer is introduced to reach the deeper layers. The initial layers are embedded on mobile devices, the intermediate layers are presented in cloudlet and the deeper layers are embedded in the cloud server. The classification of the input sensed data (image) is computed in the initial layers residing on the mobile device. Thus, the back-propagation technique is used to train the layers by evaluating all the layers until a desired confident classification result is obtained. Therefore, if it cannot classify the sample sensed data with sufficient confidence, it is then transferred to the intermediate layers in the cloudlet for the classification process. The deeper layers are only invoked when both the initial and intermediate layers are unable to classify the input data set to meet the desired confidence candidate. In addition, the CDCNN can decide whether to reject or accept classifications based on the threshold value passed as an argument at runtime. This improves the accuracy and speed of the entire classification process. Table 4 identifies the problems solved, performance results, and weaknesses of the existing algorithms used for predicting data redundancy. It also indicates the processes adopted by the algorithms, edge devices, and cloud IaaS resource components as indicated in previous literature.

5.3. Cloud resource provisioning for user requests

Providing of efficient resource allocation ensures satisfactory cloud service for end-user requests. In IoT-enabled edge cloud computing, resources are allocated as Physical Machines and Virtual Machines in the cloud IaaS platform, as shown in Fig. 7.

How to integrate virtual machines into servers to support the requested task determines the ability to minimize the resource allocation problem [83]. This research focuses on the problem of load balancing when migrating virtual machine(s) from the source server to the destination server for executing data filtering or analytical application requests. Load balancing refers to the pattern in which resources are distributed to avoid overloading any Machine (Servers and VMs) as resources are optimally utilized [84]. Also, it determines the migration of tasks to underutilized VMs and Servers for effective resource sharing [85]. In this article, we analyze the existing algorithms used for resolving the related issues of load balancing while allocating resources to execute the filtering data or analytic application requests on the cloud IaaS platform.

Jing et al. [86] proposed a Dynamic Priority and Load Balancing (DPLB) algorithm for VMs resource(s) load balancing carrying the scheduling of IoT application request tasks execution on IaaS. The dynamic priority function is composed of task value density and task computation urgency. In addition, the priority is subsequently increased over a period of time to ensure timely execution of each task on the queue. The scheduling function consists of Earliest Completion Time (ECT) and retrieving the load status information of each VM with the support of publish/subscribe method. The task are ordered according to their priority level, and the tasks with highest priority are scheduled first to the optimal VMs among heterogeneous VMs that meet the QoS requirements with the support of the task migration manager. The Brier Score method is used to predict an overloaded VM, whereby if a VM

Table 4
Comparison of redundant data elimination techniques.

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
SVM–T-RFE: A novel gene selection algorithm for identifying metastasis-related genes in colorectal cancer using gene expression profiles [53]	Support Vector Machine (SVM) Recursive Function Elimination	Clustering	Inefficient elimination of feature redundancy	Efficiently eliminated redundant data with minimum computation time	Candidate feature set consists of highly correlated features	PC	N/A	N/A	N/A
Feature selection and analysis on correlated gas sensor data with recursive feature elimination [54]	SVM Recursive Function Elimination-based Correlation Bias Reduction (SVM-RFE+CBR)	Clustering	Candidate feature set consists of highly correlated features	Improved elimination of feature redundancy while retrieving actual sensed data	N/S	PC	Yes	N/S	N/S
On reliability of neural network sensitivity analysis applied for sensor array optimization [55]	Neural Network Sensitivity (NNS)	Deep learning	Inappropriate selection of desired features among various features	Effectively and efficiently retrieved the best features with improved accuracy	Unclear result due to limited number of input features for training features	PC	N/A	N/A	N/A
Sensor array optimization for mobile electronic nose: wavelet transform and filter based feature selection approach [56]	Fast Correlation-based Filter (FCBF) algorithm	Classification	Unclear result due to limited number of input features and overlapping of features selectivity	Obtained best combination of features while discarding redundant ones	Computation time complexity	Remote Server	Yes	N/S	N/S
Fractional-order embedding multi-set canonical correlations with applications to multi-feature fusion and recognition [57]	Fractional-order Embedding Multiset Canonical Correlations (FEMCCs)	Classification	Deviation of relevant sensing data due to noise and limited training samples	Effectiveness and robustness in eliminating noisy data	Not considering vital correlation among different feature sets	Server	N/A	N/A	N/A
Discriminant correlation analysis: real-time feature level fusion for multimodal biometric recognition [58]	Discriminant Correlation Analysis (DCA)	Classification	The identification and elimination of redundant feature between-class feature similarities	Improved accuracy for detecting and elimination of redundant features	Still pose with feature redundancy within the intra and extra class in multiple classes or a single class	Laptop PC	N/A	N/A	N/A
Enhanced feature fusion through irrelevant redundancy elimination in intra-class and extra-class discriminative correlation analysis [59]	Intra-class and Extra-class Discriminative Correlation Analysis (IEDCA-IRE)	Classification	The neglecting of some correlation information among various feature sets due to over-fitting between data points	Improved accuracy of detection and elimination of feature redundancy	Computation time complexity	Remote Server	Yes	N/S	N/S
Mobile-cloud assisted video summarization framework for	Jeffry-divergence Boolean Series-	Classification	Issue of duplicate sensed images	Improved accuracy of relevant sensed image retrieval	Constrained with computation time complexity and cannot be applied	Mobile Phones	N/A	N/A	N/A

(continued on next page)

Table 4 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
efficient Management of remote sensing data [60]	based on Ensemble SVM			while discarding irrelevant ones	for sequence sensed dataset				
IoT as a applications: cloud-based building management system for the internet of things [61]	Correlation Feature Selection-based heuristic algorithm	Classification	Computation time complexity for optimal feature selection	Minimized dimensionality of sensed data sets with less execution time	Cannot be used for time-series sensing data	Local Server	Yes	1	N/S
Research on the fusion method of spatial data and multimedia information of multimedia sensor networks in cloud computing environment [62]	Scale Invariant Feature Transform algorithm	Classification	Problem of extracting actual sensed data from massive sensed data sets	Minimized computation resource usage while enhancing accuracy of extracting actual sensed data	Unable to consider the spatial correlations among sensed data set	Controller Raspberry Pi	Yes	N/S	N/S
A cloud-based monitoring system via face recognition using Gabor and CS-LBP features [63]	Center-symmetric Local Gabor Binary Pattern algorithm	Classification	Problem of poor facial images and the complexity of Gabor filter	Reduced the Gabor filter complexity and improved rotational invariance	Consumes computation resources	Desktop computer	Yes	N/S	N/S
A big data classification approach using LDA with an enhanced SVM method for ECG signals in cloud computing [64]	Linear Discriminant Analysis-based Enhanced Support Vector algorithm	Classification	Error during classification of sensed data for the retrieval of relevant ones	Improved the sensitivity and specificity and reduced the error during classification	Highly computational intensive	Mobile phones	Yes	N/S	N/S
An incremental CFS algorithm for clustering large data in industrial internet of things [65]	Incremental Fast Searching Clustering based K-Mediods	Clustering	Problem of clustering dense peaks of dynamic sensory data	Improved the effectiveness of clustering accuracy with minimum computation time, compared to other methods	Computation time consuming when all the clusters are to be merged.	N/S	Yes	10	N/S
BEATS: Block of Eigen-values Algorithm for Time Series Segmentation [66]	Block of Eigen-values algorithm	Clustering	Unexpected drift data points in big data set	Efficient detection of drifts with an improved classification and clustering accuracy	Unable to estimate the block size before data arrival and involves computation time complexity	Local Server	Yes	N/S	N/S
An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT [67]	Efficient High-order Tensor Fuzzy C-means algorithm	Clustering	The inability of fuzzy c-means algorithm to cluster big sensing data stream in low end IoT devices such as controllers and mobile phones.	Improved computation efficiency in terms of fo r timeliness and significant level of clustering accuracy as compared to the conventional method	There is still limitation in the aspect of clustering accuracy as it mainly focuses on the minimum usage of computation resources	Remote Server	N/A	N/A	N/A
Social choice considerations in cloud-assisted WBAN architecture for post-disaster healthcare:	Banag Pseudo-cluster based aggregation	Clustering	Problem of sensed data filtering from a data set	Improved reliability probability based on aggregation of sensor data in terms of their level of need	Weakness in aggregation sensed data from heterogeneous (big data) sources	Broker server	Yes	N/S	N/S

(*continued on next page*)

Table 4 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
data aggregation and channelization [68]									
Federated internet of things and cloud computing pervasive patient health monitoring system [69]	Cobweb Expectation Maximization-based K-Means	Clustering	Highly dimensionality of sensed data set and its noisy nature	Improved the quality of sensed data by reducing its dimensionality based on aggregation strategy	Computationally intensive and consumes a large amount of computer memory	Mobile Phone	Yes	N/S	N/S
A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure [70]	Two-step K-means Clustering Algorithm	Clustering	Numerous blurry, background images (useless) data that limits classification accuracy and delayed transmission of the data to the cloud	Eliminated unusable sensor data resulting in improved clustering accuracy	Unable to discover the actual correlation among the discovered patterns in the sensed dataset	Mobile Phone	Yes	N/S	N/S
Predictive analytics for complex IoT data streams [71]	Adaptive Moving Window Regression algorithm	Regression	Challenge of complex event streaming data without leveraging historical data for prediction	Improved prediction accuracy in near real-time and minimized the computation complexity	Utilizes huge amount of computation resource (memory space)	N/A	Yes	N/S	N/S
Effective features to classify big data using social internet of things [72]	Elephant Herd Optimization-based Linear Kernel Support Vector (EHO-LKSV) algorithm	Optimization/ Classification	Delay in the computation processing of sensed data feature selection	Enhanced feature selection accuracy with minimum computation time and memory usage	N/S	Fog Sever	N/A	N/A	N/A
A novel data reduction technique with fault-tolerance for internet-of-things [73]	Perceptually Important Points (PIP)	Classification	Problem of both local and global optima in sensed data reduction.	Effective and efficient elimination of duplicate sensed data with same time retrieval	Eliminate relevant sensed data alongside with duplicates ones due to missing data	Primary and Secondary Server	Yes	N/S	N/S
Toward modeling and optimization of features selection in big data-based social internet of things [74]	Hadoop Artificial Bee Colony algorithm	Optimization	Computational complexity involves extracting of features in real-time IoT streaming data	Improved feature selection accuracy with response to timeliness	Not Specified	Multi-cluster Hadoop with i5 3.4 GHz and 8 GB RAM	Yes	N/S	N/S
A novel deep learning method for aircraft landing speed prediction based on cloud-based sensor data [75]	Deep Learning Long-short term memory (LSTM) algorithm	Deep learning	Inaccurate classification of sensed data retrieved from aircraft to determine the safety of its landing speed	Improved classification accuracy to some extent in a timely manner	Weakness in the selection of optimal parameters to determine relevant sensed data from irrelevant ones.	N/A	Yes	N/S	N/S
Deep reinforcement learning in support of IoT and smart city services [76]	Deep Reinforcement Learning algorithm	Deep learning	Problem of close estimation of the target locations in an indoor environment	Improved classification accuracy and performance of locating target objects	Highly computationally complex	Fog server	N/A	N/A	N/A

(*continued on next page*)

Table 4 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	Weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
Big data analytics for prediction of remaining useful life based on deep learning [77]	Integrated Deep Auto-Encoder algorithm	Deep learning	Ineffective retrieval of desired sensed data from massive data set for prediction purpose	Effective extraction of desired sensed data which enhances prediction accuracy	Computationally intensive	Fog server	Yes	N/A	N/A
Learning IoT in edge: deep learning for the internet of things with edge computing [78]	Deep learning algorithm	Deep learning	Complications in obtaining optimal sensed data at reduced sizes.	Improved accuracy of desired sensed data retrieval at minimum size	Highly computationally complex	Edge Servers	Yes	N/S	N/S
A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification [79]	Hybrid Multilayer Perceptron Convolution Neural Network algorithm	Deep learning	Problem of inaccurate classification of different fine spatial resolution remotely sensed images	Improved classification accuracy	Computation intensive and huge memory space usage during processing	Local Server	Yes	N/A	N/A
A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure [80]	Convolutional Neural Network algorithm	Deep learning	Inaccurate classification of sensed data and delayed transmission of the data to the cloud	Improved classification accuracy by eliminating redundant data	Classification accuracy still needs to be enhanced	Mobile Phones	Yes	N/S	N/S
Deep convolutional computation model for feature learning on big data in the internet of things [81]	Deep Convolutional Computation model algorithm	Deep learning	Inefficient detection of the correlations between heterogonous sensed data feature space	Improved classification accuracy	Highly computationally intensive	Local Sever	N/A	N/A	N/A
The cascading neural network: building the internet of smart things [82]	Cascading Deep Convolution Neural Network algorithm	Deep learning	Limited computational processing resources on embedded mobile devices	Reduced computation cost at reasonable classification accuracy in a timely manner	Classification accuracy still needs to be enhanced with an optimization algorithm	Raspberry Pi Mobile phone Cloudlet server	Yes	N/S	N/S

workload exceeds the Brier Score it is considered as overloaded, but if it falls below the Brier score it is considered as underloaded. The Task Migration Manager (TMM) then assigns or facilitates the migration of tasks to the underload VMs to balance the loads on the available VMs.

Quasi-real-time Optimization-based Adaptive SERAC3 resource allocation algorithm is introduced in Ref. [87], for selecting appropriate configuration of virtual machines to process IoT sensory big data filtering application requests on the cloud IaaS upon arrival. It solved the prevailing problem of the CP-BO algorithm, by extracting representative workloads for incoming sensing data, analyzing the data, and intelligently determining an optimal configuration (type of virtual machines, size of the virtual machine, and the number of virtual machines) for the clustering of each job in real time without considering the load balancing in PHs and VMs. However, problem of load balancing is solved in Ref. [88] by using a Virtual Machine and Selection algorithm for the processing of sensory data filtering or analytic application requests (jobs) in the cloud IaaS platform. It uses parameters such as CPU utilization, memory utilization, and job arrival rate to cluster servers into eight groups. Servers with optimal computation resources that is based on the parameters are selected from the resource pool to host VMs. As a result, the virtual machines from overloaded servers are moved to the optimal server to process new jobs as they arrive.

In [89], a Fuzzy Markov Normal (FMN) algorithm is proposed selecting VMs to be transferred from congested servers (hosts) to avoid oversubscribed hosts and minimize energy consumption. It categorizes the attributes of VMs based on their current utilization level and the workload status of the host in which they reside with the support of fuzzy logic method. It then uses the Markov Normal technique is deployed to determine which category of VMs should be migrated from the overloaded host to the less load target host. However, FMN only performs migration of VMs based on host utilization without considering the “memory utilization of VMs selection process which is the basic requirement to be established before VMs migration” [90]. Therefore, an approximation Algorithm is proposed in Ref. [91], to solve the content-based memory problems of VM selection from source to destination, with a single overloaded host and a destination host when the

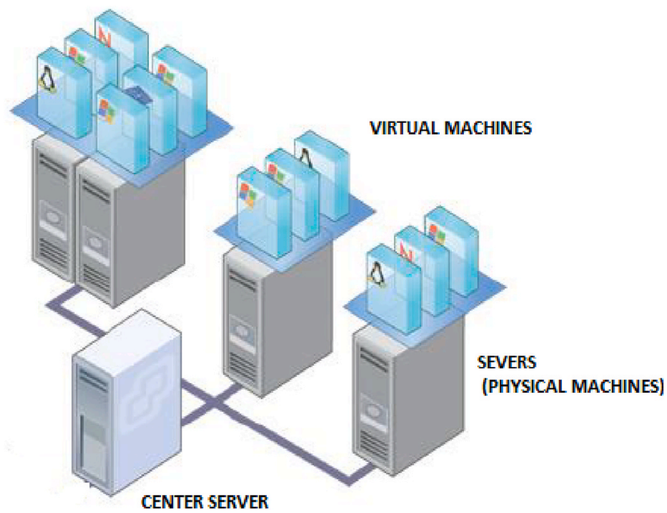

Fig. 7. An Example of servers and virtual machine(s) architecture.

overloaded threshold is fixed. It uses the memory sharing-aware placement system to exploit the content similarity of the VMs. This is done by sharing the same pages or sub-pages to simultaneously dispatch a batch of VMs at the same time from the overloaded source host and determine the appropriate destination host from various hosts that can accommodate the migrated VM.

Genetic and Particle Swarm Optimization (PSO) algorithms are proposed in Ref. [92] to optimize the selection of virtual machines, which leads to optimal utilization computational resources on the cloud IaaS platform. The GA updates the optimal selection of VMs for executing the IoT application requests, by assuming that there are a given number of VMs (chromosomes) as the possible capacity to be allocated for executing the jobs. It then calculates their fitness by using the parameters namely, CPU utilization, Turnaround, and Waiting time to determine the optimal VMs with less execution time. Otherwise, the selection operation (crossover and mutation) is repeated to generate new VMs. The selection operation is used to select and obtain a matching pair of two VMs. It also compares the two VMs and randomly generates the best one. It then implements the two-point crossover operation between two VMs to obtain two different offspring. Changes in the VMs are maintained by using the flip bit mutation, which in turn provides updates to one or more gene values in the VMs (chromosomes) from their initial state. The PSO algorithm assumes that there are numerous particles that represent VMs and in the given time of iterations. Therefore, each VM in the cloud (s) is considered as a possible solution that can be assigned to execute the incoming filtering application requests or jobs. It then computes the fitness of each VM with its g-best. If the current value is better than the p-best, it places the current location as the p-best location. On the other hand, if the current value is better than the g-best, the g-best is reformulated to the current index in the array. Thus, the optimal VM is assigned as the g-best and update the entire process for incoming job requests is updated. Experiment have been conducted to determine the performance of both algorithms, and the result shows that PSO performs better than GA.

Narman et al. [93] introduced a Dynamic Dedicated Server Scheduling (DDSS) algorithm, based on homogenous and heterogeneous servers for the processing of IoT application requests. It continuously updates the number of dedicated PMs with VMs based on requests arrival rates and their priority levels. Dedicated PMs are dynamically assigned to application requests by considering four important parameters: task arrival rate, task priority levels, total service rate of servers in the systems, and total service rate of servers capable of executing a single type of request. At the initial stage, IoT application request tasks are classified based on their arrival rate and priority levels. Then, PMs of different groups are assigned to different request classes, and the formulation of VMs into dedicated PMs for request processing is required. The number of assigned PMs is frequently updated for each class of request tasks until all tasks are fully executed.

A sub-optimal resource-based Support Vector Regression-Genetic (SVR-GA) algorithm is developed for the provisioning of cloud resources for executing application requests [94]. The SVR is responsible for predicting the resource specifications for jobs and creating the VMs to resolve the uncertainty of job arrival on real-time basis. It also evaluates the number of resources with the support of two lookup tables that keeps records of all related resource utilization rates for each VMs. It also determines whether the VM should be increased or decreased based on the number of application request tasks. The genetic algorithm is then used to assign VMs to PHs for job execution. It adjusts itself optimally to allocate VMs for the new arrivals. Jeyarani et al. [95], developed an Adaptive Power-aware Virtual Machine Provisionary (APA-VMP) that efficiently allocates VMs to a group of servers by satisfying the specifications of an optimal number of workloads. At the initial stage, the workload of application service requests is estimated after which it allocates the desired number of VMs to active servers that can perform the job. Also, Hieu et al. [96] proposed a Max-BRU algorithm to maximize resource utilization and balance the resource across different dimensions, to reduce the total number of servers in the active state. First, a group of servers is instantiated as empty, while the initial value of servers under the running state is set to zero. Therefore, VMs are assigned to servers until all VM requests are fully assigned. Then, the average resource usage and the resource balance of all the active servers are calculated. In this way, the most suitable server is selected from the group of the existing active servers and a possible VM request is selected.

The SVR-GA, APA-VMP, and Max-BRU algorithms are prone to some limitations, which include the inability to preserve the unbalanced resources when a server reaches to its maximum computation limit, its disk and memory resources are wasted due to insufficient resources resulting in high energy consumption, the inability to achieve quality of service delivery due to real-time VM migrations and none of them are yet to be applied to IoT application requests services. However, Mekala and Viswanathan [97] address most of the above limitations, by developing an anergy-efficient resource ranking and utilization factor-based virtual machine selection (ERVS) algorithm. The algorithm is used to solve the problem of energy utilization of server and virtual machine resources for job execution. It evaluates the resource utilization rate of the jobs and properly categorizes the jobs. Then, they are assigned to the appropriate VMs that can execute each class of jobs (IoT sensed data analytics) by considering their resource utilization rate. This is realized by sorting out the highly loaded servers with the support of the Compressive Resource Ranking (CRB) scheme, which places more emphasis on resource utilization and energy consumption of servers. Then, VMs are assigned to execute the jobs (IoT sensory data filtering or analytic application requests), by considering a limited type of job with deadlines and the resource requirements for executing the specified job.

A hybrid-based Combinatorial Ordering First-Fit Genetic (COFFGA) and Combinatorial Ordering Next Fit Genetic (CONFGA) algorithms are developed in Ref. [98], to reduce the resource waste per server and the total number of servers in an active state. It determines the optimal VMs that are capable of executing the requested workloads to be migrated to the desired servers that are in an active state. While the First and Next Fit heuristic techniques are responsible for making migration decisions to reduce the total resource waste in each of the physical servers that are in an active state, and the number of non-ideal physical servers. However, the hybrid-based algorithms are designed to solve the local optima problems without considering the global optima problems. Therefore, in their research, Mohiuddin and Almogren [99] solve the global optima problem by introducing a Workload Aware Virtual Machine Consolidation (WAVMCM) algorithm to switch the idle physical servers into hibernation mode. The resources of the server are classified into four classes with different resource capacities to execute different VM requests. At the initial stage, new VM requests for executing jobs are classified according the amount of resource demand after which they are assigned to the VM class that is capable of executing each job. Thus, VMs are migrated from low-load servers to intermediate-load servers within the same class. It also determines which physical servers that are inactive mode and put them into hibernation mode to minimize power consumption.

Abed and Younis [100] developed an Adaptive Firefly-enabled Weighted Round Robin (AFF-WRR) algorithm for dynamic and static load balancing on VMs to process IoT application requests. The WRR is responsible for estimating the weights of each VM based on three parameters namely CPU, memory, and latency. VMs with higher weights are considered the most viable for executing large jobs followed by the least weighted VMs. The Adaptive Firefly (AF) tracks the status of VMs and sorts them according to their weighted level. VMs with optimal resources are selected to execute incoming jobs on a real-time basis. The status of VMs is regularly monitored in milliseconds by the AF, while WRR frequently rebalances the status of VMs based on the Firefly results.

Chen and Chen [101] addressed the issue of load balancing on VMs and servers in the cloud by developing a service-oriented Virtual Machine (VM) placement algorithm. It uses the genetic algorithm to optimize the configuration of different VMs in order to achieve minimum communication overhead and total power consumption. In the initial stage, the population chromosome is generated, which represents the VMs.

It then assigns the required VMs that are capable of executing the jobs to the servers, ensuring that the VM load does not exceed the server limit. This is done through the fitness function where the communication cost between the VMs is computed and summed up to obtain the fitness value of one server. Therefore, the server with the highest fitness is randomly selected from multiple servers to execute the job. Table 5 shows the solved problems, performances, and weaknesses of the algorithms used for Cloud IaaS resource allocation for the execution of sensory data filtering or analytic application requests on IoT-based edge cloud infrastructure. It also shows the processes used by the algorithms, edge devices, and cloud data center resource components as depicted in previous research.

Basu et al. [102] introduced a hybrid Genetic-Ant Colony Optimization (GAACO) algorithm for scheduling the task requests of multiprocessor IoT applications on the Cloud IaaS. Each task is scheduled to a single processor at a time in a heterogonous processor system. A task can only be executed when its predecessors have finished execution. Simply put, once a task starts processing on a specific processor, the next task request scheduled on the same processor must wait for the previous task to finish executing. At the initial stage, the task and processor with the best fitness solution are determined among multiple processors and incoming task requests with the support of GAACO. After which the heuristic function is used to estimate the makespan (maximum execution time) taken for each task it traverses all the levels in the graph structure. Therefore, a task with a larger makespan is scheduled first in GAACO to avoid starvation processing resources. The capability of the processors is computed by the heuristic function, where the processors with the highest probabilistic ratio of resources are selected to execute the task with the highest makespan. This process is repeated for several iterations until all tasks in the graph structure are fully executed.

6. Processes and network protocols for IoT-edge cloud computing

Processes are a set of instructions that are currently being executed. These sets of instructions that are processed logically to solve specific problems which scientists call algorithms. In simple terms, processes are a set of instructions that are systematically applied by an algorithm to solve a particular problem. On the other hand, network communication protocols govern the interaction between IoT sensing devices and edge-cloud platforms. Therefore, it is important for IoT low-power devices to use appropriate communication protocols to effectively communicate with other devices and networks on the IoT-based edge cloud infrastructure.

6.1. Processes adopted in existing research

In this subsection, we discuss and analyze the processes adopted by the existing algorithms (discussed in the previous section) to solve the problems (as highlighted in the tables of section 5) on IoT-enabled edge-cloud computing.

The classification process is a supervised machine learning technique that assumes some prior knowledge to guide the partitioning operation, formulating a set of classifiers for the representation of the best distribution of patterns [103]. Furthermore, classification processes are designed to use both labeled and unlabeled data during the classification process. The set of labeled data is mainly used to train the classifier, such as the prediction function, while the unlabeled data is classified by the classifier. The classification output is a finite set of predefined discrete classes or values, depending on the number of classes the classification problems belong to either binary or multi-class categories [104]. The binary category or classification consists of two labels e.g. 0/1, good/bad, and white/black, while the multi-class category consists of multiple labels. Consequently, the quality of the classification results is verified by determining the number of test patterns that are allocated to the corresponding collections, which is called the accuracy rate.

The regression process is used to design the correlation between input and output variables to achieve a predictive solution. The result of regression processes is determine in the continuous domain. For example, in a diabetic monitoring application, a regression can predict the symptoms of diabetes based on previous information. In general, the regression allows the prediction of the outcome of a specific event. It is widely used in the updating of IoT health and agriculture application domains.

The clustering process is an unsupervised learning process that extracts hidden patterns and structures from a given data set. Unlike classification which has some prior knowledge to strategize the partitioning operation, clustering has no pre-knowledge of the strategy to be used for the extraction process. It aggregates the data into groups, based on their similar features and common structure as well as the data points in different dissimilar clusters. Clustering is mainly used in recommender systems and outlier detection. The verification and evaluation of clustering results is based on the amount or number of dimensions of the data set to which the clustering algorithm is applied. For example, the sum of squared errors is mainly used for data clustering while the peak-signal-to-noise ratio is mainly used for image clustering [105].

Deep learning is a machine learning technique that consists of deep and complex architectures [106,107]. These architectures consist of many layers that convert input (e.g. images) into output data (e.g. an actual image) while learning progressively on higher-level features [108]. Deep leering, also known as Deep Neural Networks (DNN), was considered complex to train data effectively and efficiently, it performs both classification and clustering processes during operation. It began to gain popularity in 2010 when it was discovered that training and analysis of large, high-dimension IoT big data could be realized with optimal results [109]. The stacked auto-encoders (SAEs) and DNN layers sequentially in an unsupervised manner (pre-training), and fine-tuning the stacked network with a supervised approach, could provide better performance. However, they are known to be inflexible and require a reasonable amount of work to generate acceptable results.

Optimization is the process of modifying some features of a system to improve its performance or use limited resources more efficiently. For example, an algorithm can be optimized to speed up its process execution faster or to use minimum memory resources during process execution. Optimization techniques are mainly based on a bio-inspired model whose algorithms are mainly used to solve optimization problems. The optimization-based process is adopted by the algorithms in

Table 5
Comparison of resource allocation techniques for executing IoT applications.

Article Title	Algorithm	Process	Problem Resolve	Outcome	weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
An open scheduling framework for QoS resource management in the internet of things [86]	Dynamic Priority and Load Balancing (DPLB) algorithm	Optimization	Inability to execute dependent tasks and violation of SLA	Reduced makespan and improved load balance on VMs	Local and global optimum issue	N/A	Yes	12	124
SERAC3: Smart and economical resource allocation for big data clusters in community clouds [87]	Quasi-real-time Optimization based Adaptive SERAC3 resource allocation algorithm	Optimization	Exhaustive search cost for optimal resource selection.	Improved the selection of optimal configurations with lower exhaustive search cost	High resource utilization due to inefficient load balancing	N/S	Yes	16	N/S
Resource-aware virtual machine migration in IoT cloud [88]	Resource-aware Virtual Machine and Selection algorithm	Optimization	Issue of unbalanced load due to unforeseen changes upon job arrival	Reduced the dispatch time for the provisioning of PHs and VMs in the cloud data center	Unable to consider the bandwidth communication between VMs	Raspberry Pi B+	Yes	3	12
Improvement of energy efficiency at cloud data center based on fuzzy Markov normal algorithm VM selection in dynamic VM consolidation [89]	Fuzzy Markov Normal Algorithm	Clustering	Inefficient selection of VMs migration from overloaded host	Improved Load balancing with optimal placement of VMs on target servers and minimal energy consumption	Unable to consider the VMs memory contents before migration	N/S	Yes	16	640
An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud [91]	Approximation Algorithm	Optimization	Latency delay of VMs dispatched from overloaded to destination server	Reduced the migrated VMs memory data with minimum energy consumption	Energy consumption is still on the high side	N/A	Yes	100	4000
A hybrid model of Internet of Things and cloud computing to manage big data in health services applications [92]	Particle Swarm Optimizer algorithm Genetic Algorithm	Optimization	Global optima entrapment and tasks computation time complexity	Reduced computation time and optimal provisioning of storage	Weakness in local space entrapment	Router	Yes	100	1000
Scheduling internet of things applications in cloud computing [93]	Dynamic Dedicated server scheduling algorithm	Optimization	Inefficient provisioning of servers for homogenous and heterogeneous IoT data	Minimized computation delay and improved utilization of servers	Weakness in load balancing among servers	Not Specified	Yes	8	N/S
An adaptive resource management scheme in cloud computing [94]	Support Vector Regression-Genetic (SVR-GA) algorithm	Regression/ Optimization	SLA variation for resource utilization	Improved resource utilization configurations with SLA between VMs and cloud service Providers	Not considering computation cost	N/A	Yes	6	100
An adaptive Resource management scheme in cloud computing [95]	Adaptive Power-aware Virtual Machine Provisioner	Optimization	Unexpected overload and high energy consumption	Improved load balancing with less energy utilization	Still challenged with high energy consumption	N/A	Yes	100	N/S

(continued on next page)

Table 5 (*continued*)

Article Title	Algorithm	Process	Problem Resolve	Outcome	weakness	Edge Device	Cloud Data Center		
							Cloud Storage Server	No. of Physical Machine (PMs)	No. of Virtual Machine (Vms)
	(APA-VMP) algorithm								
A virtual machine placement algorithm for balanced resource utilization in cloud data centers [96]	Max-BRU algorithm	Optimization	Unbalanced load due to inefficient activation of desired servers	Improved and balanced use of resources of multiple types of servers deployed	Unable to estimate overloaded PMs upon arrival of new jobs.	N/A	Yes	150	N/S
Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for IoT [97]	Energy-efficient resource ranking and utilization factor-based virtual machine selection algorithm	Optimization	Unbalanced resource utilization and high energy consumption	Improved the utilization rate and minimize the number of live VM migrations with less energy consumption	Weakness in local search entrapment and computation time complexity	Laptop Pc	Yes	100	500
Multi-capacity combinatorial ordering GA in application to Cloud resources allocation and efficient virtual machines consolidation [98]	Combinatorial Ordering First-Fit Genetic and Combinatorial Ordering Next Fit Genetic algorithms	Optimization	High number of running servers and resource waste per server in local search space	Minimized the total number of running servers with less resource waste	Unable to consider the issue of global optima while determining the best VMs among various ones		Yes	128	340
Workload aware VM consolidation method in edge/cloud computing for IoT applications [99]	Workload Aware Virtual Machine Consolidation algorithm	Optimization	Inability for edge cloud data centers to process tasks in a power-saving mode and the issue of global entrapment	Improved convergence rate with minimum active server usage and less energy consumption	Not considering the communication overhead between servers and VMs	Laptop Pc	Yes	500	1500
Developing load balancing for IoT-cloud computing based on Advanced Firefly and weighted round Robin algorithms [100]	Firefly and Weighted Round Robin algorithms	Optimization	Overloaded PMs due to unbalanced load on every resource	Improved resource utilization with minimum response time	Inefficient searching of candidate resources for job execution		Yes	1000	5000
Service oriented cloud VM placement strategy for internet of things [101]	Service-oriented virtual machine placement algorithm	Optimization	Challenges of high communication overhead between VMs under the same service	Minimized communication cost between VMs, energy usage and the total PM utility	Unable to schedule the VMs for task execution which disrupt load balancing in the PMs		Yes	250	N/S
An intelligent /cognitive model of task scheduling for IoT applications In cloud computing environment [102]	Hybrid Genetic-Ant Colony Optimization (GAACO) algorithm	Optimization	Scheduling task dependency	Efficient load balancing with reduced makespan	Not considering local search entrapment		Yes	1000	2000

previous research, to solve optimization problems for the allocation of resources required for the execution of IoT data filtering (outliers and redundancy elimination) on analytic applications have been extensively analyzed in this paper.

6.2. Network communication protocols deployed in IoT-Edge Cloud computing

Communication protocols such as message Query Telemetry Transfer (MQTT), Wireless Fidelity (WiFi), Bluetooth, General Packet Radio

Service (GPRS), and Advanced Message Queue (AMQP) were used in previous research which is briefly discussed as follows:

Message Query Telemetry Transfer (MQTT) was invented by IBM in the year 1999 as a standardized publish/subscribe push protocol. It is specifically designed to facilitate the transmission of data under long network delays and low-bandwidth network conditions [110,111]. It mainly runs on both TCP/IP and other network protocol that is designed to provide lossless and bidirectional connection. Consequently, MQTT is suitable for resource-constrained IoT sensing devices that uses unreliable or limited bandwidth channels [112]. It was standardized at Oasis in 2013 with a channel bandwidth of 5–20 MHz, Downlink rate of 256 MB and an uplink rate of 127 MB over the TCP/IP port of 8883.

Bluetooth is a wireless communication protocol designed to provide short-range connectivity for small devices such as smartphones, laptops, and hand-held devices. It was standardized by the 802.15.3 in 1999, and operates in the 2.4 GHz frequency band at a low rate of 200 kb/s. Its main function is to allow audio and data streaming between devices. However, it consumes power energy during data transmission between devices. This led to the introduction of Bluetooth Low Energy (BLE) in the year 2010 to address this high power consumption. BLE is designed to extend the application of Bluetooth for use in low-power devices such as wireless sensors and wireless controllers [113]. Currently, the IETF 6LoWPAN Working Group (WG) has already recognized the importance of BLE for the Internet of Things and is beginning to develop a specification for the transmission of IPv6 packets over BLE [114,115]. It is most commonly used by IoT sensing devices to transmit data to other devices.

Fourth/Fifth Generation (4G/5G)-LTE Fourth Generation- Long-Term Evolution (4G LTE) are wireless network protocols designed and deployed for the Internet Protocol (IP)-based services, such as the combination of multimedia capabilities and applications that with high-speed mobile broadband [116]. It is considered to be ten times faster than 3G in terms of transmission speed and covers a wider range. As a result, its Packet Core (EPC) and IP-based network framework, enable the smooth delivery of voice and data packets as compared to the older models of cell towers using GSM and UMTS. However, it is fast reaching its limits due to the increasing demand for wireless data transfer as the use of mobile phone usage grows and the reduction of latency in end-to-end connections due to the physical imposition of the Internet. Therefore, the Fifth Generation (5G) mobile protocol has been introduced to solve the aforementioned issues of the 4G. 5G is specifically designed to support efficiently support massive machine-to-machine and critical communications. Thus, a large number of actors and sensors/meters that are deployed anywhere in the landscape will be able to transmit their sensed data to other devices with a very low response time and high reliability [117]. It also has the potential to provide mobile broadband services such as high-speed multimedia streaming, video-conferencing, Internet browsing, Voice-over-IP (VoIP), and efficient downloading and uploading of large files.

Advanced Message Queue (AMQP) is a protocol that originated in the financial sector. It has been standardized by Oasis as a ubiquitous, secure reliable, and open Internet protocol for handling messages [118]. It is regarded as a messaging middleware that uses different transport protocols. AMQP provides asynchronous publish/subscribe communication with messaging, in addition to its store-and-forward feature that ensures reliability during and after network disruptions [119,120]. This means that AMQP has the potential to be used in hazardous or hostile environments, as long as the overhead is not very high.

Wireless Infidelity (Wi-Fi) is used to connect wireless devices such as laptops, smartphones, and PDAs. It is a brand of wireless communication technology that is held by the Wi-Fi alliance to improve the interoperability between wireless networking products based on the IEEE802.11 standard [121]. It has a coverage range of 46 m (indoor) and 100 m (outdoor) with a bandwidth channel of 20–40 MHz, followed by a downlink rate of 600Mbps and an uplink of 248Mbps at a frequency band of 2.4 GHz.

7. Potential challenges of IoT-enabled cloud computing infrastructure

Whiel IoT-enabled cloud systems tend to solve many problems, there are a reasonable number of challenges that have yet to be addressed. This is because the potential solutions needed to solve these challenges have not been unravelled by the algorithms in previous research. Also, some of these remaining challenges require a handful of consistent efforts from IoT-Cloud researchers and development communities, governments, policy makers, and platform/hardware providers. Some of these challenges are discussed as follows;

Unstructured IoT sensing data. In real-world sensing events, the sensed data generated by sensor devices is unstructured due to their dynamic and heterogeneous nature. While NoSQL and Ubuntu servers are designed to store the unstructured data, they have yet to make a significant impact on real-world IoT sensory enabled cloud infrastructure, as most researchers use structured data sources to experiment. However, the emergence of data lakes has proven to handle large volumes of IoT sensor data. It is able to store both unstructured and structured data without any predetermined idea of how data will be used. It also does not use query languages or scheme mapping and can store any type of data without limitations. Lake is challenged with two major issues. First, loss of agility may occur when it is utilized to store a huge pool of data that urgently needs analysis and decision making. Because they have to go through several processes before any meaningful data can be extracted from the data sample. Secondly, data interchange may happen in the future since any data can be stored or inserted [122]. This problem can be avoided by attaching metadata to the stored data and ensuring the attribute or source of the data. Therefore, it is necessary to further investigate on how algorithms can be used to manage these unstructured sensory data both in the simulation environment and in a real-world scenario.

Protocol diversity and Standardization. The IoT-enabled edge cloud platform is challenged with a universal protocol and standard, as different protocols are used to communicate and interact between devices of different development standards. While the platform has been designed to enable multiple protocols to work together due to different requirements and their intended uses, but may lack the potential to support multiple protocols extensively. Therefore, it is worth further exploring the development of an intelligent gateway as a possible solution that can provide seamless interoperability and integration between different protocols and algorithms that can intelligently select the optimal transmission channels for efficient data delivery. On the other hand, various organizations, such as 3GPP, IEEE, ETSI, and M2M made some significant efforts to enforce standards for the development of IoT devices. They assume that interoperability will be provided by the aforementioned standardization activities, but they may lead to higher uncertainty as they all provide specific and isolated solutions that can only cover their domains [123].

Integration of contextual information. IoT data must be integrated with other data sources, such as context information that complement the understanding of the environment [124]. This is because IoT-sensed data cannot understand the environment on its own. The emergence of algorithms tends to speed up data filtering, analysis, and efficient reasoning due to the limited search space for the reasoning engine. For example, a sensor camera with the facial recognition cability can perform surveillance in different contexts such as in government buildings and residential areas [125]. Therefore, the sensed image data collected from different contexts can assist the system to determine the optimal action to be taken based on the retrieved face of an individual.

Overloading communication networks. With a large number of IoT-enabled edge cloud components, maintenance and configuration of their underlying physical Machine-To-Machine (M2M) interactions and networks becomes more complex. The dynamicity and heterogeneity of IoT big sensing data rapidly overwhelms the communication networks of the IoT-enabled Edge Cloud platform. Therefore, the volume and speed of the data must be taken into account in order to provide optimal Quality of Service (QoS). One way to address this issue is to provide for the storage and management of IoT sensor data across tiers of the IoT-based edge cloud (ITC). This will compel application designers to deploy complete data contextualization algorithms and techniques to obtain optimal QoS delivery across the ITC platform. The contextual techniques must consider the storage capabilities of the essential processing devices such as the sensing devices, microcontrollers, and edge servers and the cloud data centers.

Security challenge in collaborative edge-cloud processing. Further research is needed on how to perform cloud-side computations to encrypt IoT sensor data, without revealing secrets or privacy to cloud service providers. In addition, how the edge can send the sensing data to the cloud in a secure manner, ensuring that the sensing data is not corrupted in the edge processing units, and cannot be intercepted by unauthorized persons (intruders) while in transit to the cloud platform, needs to be addressed.

Real-time Filtering/Analytic data. Achieving useful and intelligent information in real time from a huge volume of sensed data collected from several multiple IoT sensing devices has become a major challenge. This is due to the unavailability real-time stream mining approaches. One way to overcome this challenge is the use of edge devices, which has already been proposed. Nevertheless, there are other solutions (such as algorithmic techniques) that are in the early stages of implementation and need to be optimized to extract meaningful and intelligent data on a real-time basis, which needs to be addressed in the future.

8. General discussion and conclusion

As expected, the algorithms were able to resolve issues related to sensed data filtering based on outlier detection and redundancy elimination in a given data set. In addition, issues related to load balancing for resource allocation, such as migrating VMs from source to target server(s) to perform the execution of sensed data filtering or analytic application requests, were significantly resolved. Outliers were primarily detected by considering the data type, spatio-temporal, attributes correlations, user specification threshold, outlier score, and identifying the type of outlier (error and event). There are two main types of data, namely linear and non-linear. The linear data type is known as static and is structured sequentially either in a list(s) or frame(s) format. Non-linear is dynamic data and is also known as time series or streaming data. Spatio-temporal simply means the distances between sensing data and time upon arrival from a particular source (sensor). In other words, sensing data within a specific close range are considered normal data while others are classified as outliers or anomalies. The similarity (correlation) between several data in a given dataset is also determined, as those with the same values are either clustered or aggregated into several groups or subsets according to their similarity level. Outliers within the subsets are then identified based on threshold(s) or score.

We also observed that outliers are of two types as detected by some of the existing algorithms, namely error and event. Error outliers are generated by defect sensors which are often classified as irrelevant or un-wanted data and are therefore eliminated from the dataset. Event outliers, on the other hand, are useful data, most often used to report or predict unforeseen circumstances. For example, the detection of a gas leak from a cylinder is called an event outlier. In terms of redundancy, feature selection and pattern recognition have been strongly. Features or attributes of a given data set are subjected to a similarity check to identify data with similar attributes or features. Thus, similar features are selected to be merged into a single data feature or better still one out of the similar data features is retained while others are eliminated. Similarly, similar data patterns are classified or clustered together while the irrelevant ones are identified and discarded.

Load balancing issues have mainly been solved by considering the number of incoming requests prior to arrival while searching for optimal or under-loaded VMs to migrate from source (overloaded) to target (under-loaded) server(s) to execute jobs. This, avoid execution time complexity and overloading of available recourses (VMs and Servers). There are various servers (physical machines) in the cloud IaaS platform dedicated to specific tasks leading to the effective management of computation and storage resource provisioning. According to Ref. [14], there are three main types of sensed data servers in the cloud IaaS namely NoSQL, Relational Database (MySQL), and Hadoop servers. The NoSQL server is mainly designed to store and manage IoT sensed data due to their unstructured pattern. It has some features such as distributed storage, dynamic schema, and horizontal scalability. However, it is limited in its ability to maintain consistency, isolation, atomicity, and durability of sensed data. In addition, it partially supports distributed queries. On the other hand, Hadoop servers are unique distributed file repositories that store and efficiently manage massive unstructured data. It enables IoT sensed data to be generated in XML format.

According to Shvachko et al. [103], the combination of both NoSQL and Hadoop servers enables unified management and access to sensed data. Relational database (MySQL) server stores massive structured data. However, different data are generated rapidly and the relationship between these data is of importance for a multitenant data storage system [104]. Therefore, virtual relational data is merged with various conventional relational data in a single schema. Despite the potential features of the cloud servers, they are still prone to the massively heterogeneous and dynamic nature of IoT big data. One way to solve this problem is to use a virtual machine for effective and reliable data processing and storage management on servers. Virtual machines subsets of a server that can be used to perform highly intensive computational tasks. This enables a server to perform two or more tasks simulanously, such as providing storage space for incoming sensed data and at the same time performing data filtering or analytic operations on the sensed data using algorithms (e.g. algorithms used for both data outlier and redundancy detection) based on user application requests.

The Observation in Fig. 8(a) shows that redundancy problems were mainly handled by the classification process, followed by deep learning and clustering, with limited use of optimization and less use of regression processes. On the other hand, the clustering process was mainly used to detect outlier-related problems, followed by classification and deep learning, with limited use of regression and with no use of optimization. The optimization process happens to be the most deplorable process for resource allocation in the Cloud IaaS, to execute sensory data filtering and analytic application requests (jobs), followed by clustering with limited usage of regression and without the use of classification process. In addition, clustering seems to outperform other processes in terms of its usage by the existing algorithms studied in this research, as shown in Fig. 8(b). Followed by classification, optimization, deep learning, and regression processes respectively. This shows that the utilization of machine learning algorithms is also gaining more momentum in IoT big data filtering and analytics on IoT-enabled edge cloud computing.

Observations from the tabulated information indicated that a reasonable amount of sensed data filtering algorithms used, to solve

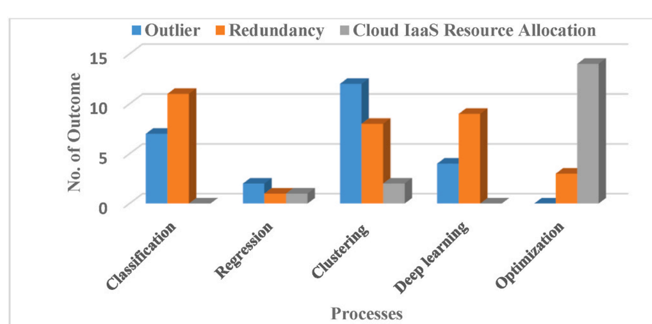

Fig. 8a. Utilization frequency of processes.

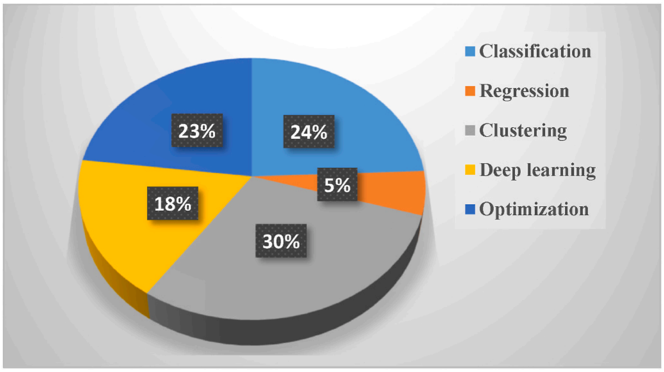

Fig. 8b. The level of Utilizing Processes in (%).

outlier and redundancy related problems were implemented at the edge and edge/cloud respectively. The processes at the edge/cloud are mainly based on retrieving relevant sensed data while discarding the irrelevant ones. In addition, the algorithms executed only at the edge platform are mainly accessed immediately by the end-user applications. Fig. 9 shows that most of the algorithms are implemented in the edge/cloud respectively. This shows that the use of cloud to exploit the limitations of IoT sensing device(s) and that of edge devices to process of IoT big data are gaining more momentum in this research area.

In the aspect of processes adopted by the existing algorithms, the clustering process outperforms other processes in terms of usability level. It can extract useful information from large sensed data as compared to others, due to its sensitive nature to outliers and redundant (noisy) sensed data. Clustering is done by partitioning based on the distance between instances, where each instance is identified as a cluster and merges the instances that are closer to one another until all instances are fused into a single cluster. Observation also shows that most of the clustering process was implemented on static sensed data retrieved from various sensor devices. However, clustering such as Moving Window Principal Component Analysis [50] and Robust Incremental Principal Component Analysis [51] algorithms were implemented on dynamic or real-time sensing data. Clustering methods are also known to be relatively scalable and enable the number of clusters to be specified in advance, such as the Recursive Principal Component Analysis [32], Adaptive K-means [35] and Distance-based Algorithm [44]. On the other hand, hierarchical clustering such as Enhanced Knowledge Granule [42], Hyperellipsoidal [46], and Incremental Fast Searching Clustering-based K-Mediods [65], specifies the number of clusters itself as it performs operation on any given dataset.

Classification methods are mainly used in health-related sensory data collection to predict redundant data, as can be seen from the existing research. They are known for their efficiency in terms of time and space complexity and are suitable for real-time sensing data. Furthermore, the classification process is easy to develop and maintain in parallel hardware such as the cloud data center. However, it requires lengthy training and testing procedures on sensed data with poor interpretability. Deep learning mainly combines the services of clustering and classification to perform its operations on sensed data. They are most suitable for large sensory data as observed from previous research and tend to achieve high accuracy in terms of performance compared to other methods. However, they require a large amount of storage space and are more time-consuming to run than the other methods. The optimization process has been mainly used at the cloud data center to improve efficiency in terms of computation time completion (makespan), minimum resource utilization and energy consumption as observed from previous algorithms. Their main objective is to prioritize available resources with optimal ability to execute the required task.

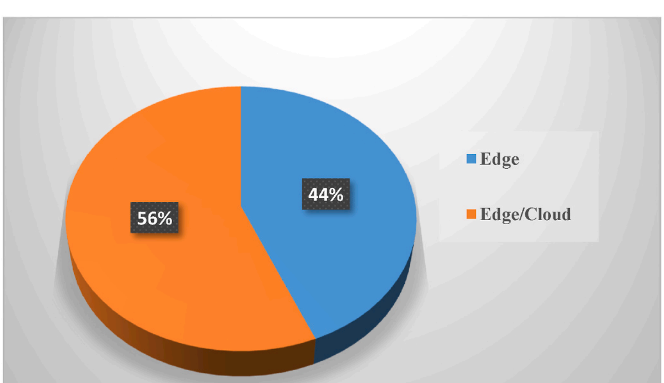

Fig. 9. Analysis of comparative usage of algorithms on Edge and Edge/Cloud.

In conclusion, data filtering or analytic algorithms are the main tools used to extract knowledge from massive data generated from various IoT sensing devices. On the other hand resource allocation algorithms are used to provide optimal computation and storage resources for executing data filtering/analytic application requests in IoT-enabled cloud IaaS platform. Therefore, to achieve the desired knowledge information, appropriate filtering algorithms that are effective and efficient need to be deployed due to the characteristic nature of IoT-sensed big data. In this paper, we identify and discuss some related literature surveys on the IoT-based edge cloud domain, which motivated the current research under study. Extensive background information about IoT devices, sensing data characteristics and factors that motivating the integration of IoT, edge/cloud. A detailed description of the adopted research methodology used to update the current research under consideration. Filtering/analytic algorithms from previous researches were analyzed based on issues related to outlier detection, redundant data discovery and elimination. The provisioning of optimal resources (PHs and VMs) for the execution of IoT application requests, taking into account load balancing issues is also presented. The problem solved, the successes and the weaknesses of algorithms are highlighted in tabular form. In addition, the processes employed by the algorithms were discussed as well as the network communication protocols used for the transmission of sensor data on the IoT-enabled edge cloud domain. Subsequently, the prevailing challenges that are yet to be resolved in the IoTenabled edge cloud infrastructure are presented to help characterize the research directions in this area. The significance of this research is to provide new insight into the discovery of event and error outliers with the use of machine and deep learning techniques. This have been ignored for long by computing research communities. The existing algorithms were applied in the healthcare sector to detect prevailing diseases and symptoms in patients and minimize cybercrimes and internet fraud. Also, in manufacturing company such as automobile production plants for detection of faulty equipment. Detection of domestic and industrial gas leak. Researchers in this area may capitalize on the weaknesses of the existing algorithms to improve their performances in future research. For example, managing IoT and cloud components to minimize energy usage and emission of carbon-dioxide. Furthermore, to improve the performance of resource allocation techniques to minimize hazardous material use and resource waste during assigned task in the cloud. Also, to apply outlier detection techniques to detect unauthorized access to data repositories and assigning resources, to protect sensitive information of cloud users' request tasks. Subsequently, optimizing the existing techniques for the retrieval of useful and intelligent data in real time will be considered in future research. The authors are currently implementing outlier techniques for detecting cancer in human brain.

Declaration of competing interest

The **authors** of the research paper titled "Survey on the Utilization of Algorithms for IoT Data Analytics on Edge-based Cloud Infrastructure", are hereby **declared** that there are no any competing or **conflict of interests** among them.

References

[1] S.K. Sharma, X. Wang, Live data analytics with collaborative edge and cloud processing in wireless IoT networks, IEEE Access 5 (2017) 4621–4635.
[2] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Comput. Network. 54 (15) (2010) 2787–2805.
[3] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of cloud computing and internet of things: a survey, Future Generat. Comput. Syst. 56 (2016) 684–700.
[4] K.M. Modieginyane, B.B. Letswamotse, R. Malekian, A.M. Abu-Mahfouz, Software defined wireless sensor networks application opportunities for efficient network management: a survey, Comput. Electr. Eng. 66 (2018) 274–287.
[5] T. Islam, S.C. Mukhopadhyay, N.K. Suryadevara, Smart sensors and internet of things: a postgraduate paper, IEEE Sensor. J. 17 (3) (2017) 577–584.
[6] C. Madhavaiah, I. Bashir, Defining cloud computing in business perspective: a review of research, METAMORPHOSIS 1 (2) (2012) 50–65.
[7] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 1–8.
[8] E.G. Petrakis, S. Sotiriadis, T. Soultanopoulos, P.T. Renta, R. Buyya, N. Bessis, Internet of Things as a Service (iTaaS): challenges and solutions for management of sensor data on the cloud and the fog, Internet of Things Journal 3 (2018) 156–174.
[9] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, EBSE Technical Report EBSE-2007-01, Durham, United Kingdom, 2007, pp. 1–53.
[10] J. Qiu, Q. Wu, G. Ding, Y. Xu, S. Feng, A survey of machine learning for big data processing, EURASIP J. Appl. Signal Process. 67 (2016) 2–16.
[11] A. Farahzadi, P. Shams, J. Rezazadeh, R. Farahbakhsh, Middleware technologies for cloud of things: a survey, Digital Communications and Networks 4 (3) (2018) 176–188.
[12] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, J. Qin, A survey on application of machine learning for internet of things, International Journal of Machine Learning and Cybernetics 9 (8) (2018) 1399–1417.
[13] M.S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A.P. Sheth, Machine learning for internet of things data analysis: a survey, Digital Communications and Networks 4 (3) (2018) 161–175.
[14] H. Cai, B. Xu, L. Jiang, A.V. Vasilakos, IoT-based big data storage systems in cloud computing: perspectives and challenges, IEEE Internet Things J. 4 (1) (2017) 75–87.
[15] S. Shadroo, A.M. Rahmani, Systematic survey of big data and data mining in internet of things, Comput. Network. 139 (2018) 19–47.
[16] M.H. Rehman, E. Ahmed, I. Yaqoob, I.A.T. Hashem, M. Imran, S. Ahmad, Big data analytics in industrial IoT using a Concentric computing model, IEEE Commun. Mag. 56 (2) (2018) 37–43.
[17] E. Ahmed, I. Yaqoob, I.A.T. Hashem, I. Khan, A.I.A. Ahmed, M. Imran, A. V. Vasilakos, The role of big data analytics in internet of things, Comput. Network. 129 (2017) 459–471.
[18] M. Ge, H. Bangui, B. Buhnova, Big data for internet of things: a survey, Future Generat. Comput. Syst. 87 (2018) 601–614.
[19] M. Aazam, S. Zeadally, K.A. Harras, Offloading in fog computing for IoT: review, enabling technologies, and research opportunities, Future Generat. Comput. Syst. 87 (2018) 278–289.
[20] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for IoT big data and streaming analytics: a survey, IEEE Communications Surveys and Tutorials 20 (4) (2018) 2923–2960.
[21] X. Fei, N. Shah, N. Verba, K.-M. Chao, V. Sanchez-Anguix, J. Lewandowski, Z. Usman, CPS data streams analytics based on machine learning for Cloud and Fog Computing: a survey, Future Generat. Comput. Syst. 90 (2019) 435–450.
[22] F. Alam, R. Mehmood, I. Katib, N.N. Albogami, A. Albeshri, Data fusion and IoT for smart ubiquitous environments: a survey, IEEE Access 5 (2017) 9533–9554.
[23] A. Ukil, S. Bandyoapdhyay, C. Puri, A. Pal, IoT healthcare analytics: the importance of anomaly detection, in: Proceedings of the 2016 IEEE International Conference on Advanced Information Networking and Applications, AINA), 2016, pp. 994–997.
[24] Mohiuddin Ahmed, A. Anwar, A.N. Mahmood, Z. Shah, M.J. Maher, An investigation of performance analysis of anomaly detection techniques for big data in scada systems, EAI Endorsed Trans. Indust. Netw. and Intelligent. Syst. 2 (3) (2015) 1–16.
[25] N. Shahid, I.H. Naqvi, S.B. Qaisar, Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey, Artif. Intell. Rev. 43 (2) (2015) 193–228.
[26] Jin Wang, S. Tang, B. Yin, X.-Y. Li, Data gathering in wireless sensor networks through intelligent compressive sensing, Proc. - IEEE INFOCOM (2012) 603–611.
[27] M. Stocker, M. Rönkkö, M. Kolehmainen, Making sense of sensor data using ontology: a discussion for residential building monitoring, in: Proceedings of IFIP International Conference on Artificial Intelligence Applications and Innovations, 2012, pp. 14–20.
[28] F. Ganz, P. Barnaghi, F. Carrez, Information abstraction for heterogeneous real world internet data, IEEE Sensor. J. 13 (10) (2013) 3793–3805.
[29] S. Kamal, R.A. Ramadan, E.-R. Fawzy, Smart outlier detection of wireless sensor network, Electronics and Energetics 29 (3) (2015) 383–393.
[30] N. Nesa, T. Ghosh, I. Banerjee, Non-parametric sequence-based learning approach for outlier detection in IoT, Future Generat. Comput. Syst. 82 (2018) 412–421.
[31] Rui Zhang, P. Ji, D. Mylaraswamy, M. Srivastava, S. Zahedi, Cooperative sensor anomaly detection using global information, Tsinghua Sci. Technol. 18 (3) (2018) 209–219.
[32] Tianqi Yu, X. Wang, A. Shami, Recursive principal component analysis-based data outlier detection and sensor data aggregation in iot systems, IEEE Internet Things J. 4 (6) (2017) 2207–2216.
[33] P.M. Kumar, U.D. Gandhi, A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases, Comput. Electr. Eng. 65 (2018) 222–235.
[34] A.F. Santamaria, F. De Rango, A. Serianni, P. Raimondo, A real IoT device deployment for eHealth applications under lightweight communication protocols, activity classifier and edge data filtering, Comput. Commun. 128 (2018) 60–73.
[35] Ş. Kolozali, D. Puschmann, M. Bermudez-Edo, P. Barnaghi, On the effect of adaptive and nonadaptive analysis of time-series sensory data, IEEE Internet Things J. 3 (6) (2016) 1084–1098.
[36] Daniel Puschmann, P. Barnaghi, R. Tafazolli, Adaptive clustering for dynamic IoT data streams, IEEE Internet Things J. 4 (1) (2017) 64–74.
[37] J. Diaz-Rozo, C. Bielza, P. Larrañaga, Clustering of data streams with dynamic Gaussian mixture models: an IoT application in industrial processes, IEEE Internet Things J. 5 (5) (2018) 3533–3547.
[38] Z. Ali, G. Muhammad, M.F. Alhamid, An automatic health monitoring system for patients suffering from voice complications in smart cities, IEEE Access 5 (2017) 3900–3908.
[39] G. Muhammad, M.F. Alhamid, M. Alsulaiman, B. Gupta, Edge computing with cloud for voice disorder assessment and treatment, IEEE Commun. Mag. 56 (4) (2018) 60–65.
[40] P. Verma, S.K. Sood, Fog assisted-IoT enabled patient health monitoring in smart homes, IEEE Internet Things J. 5 (3) (2018) 1789–1796.
[41] M.Z. Wu, Y.T. Wang, Z.C. Liao, A New Shelf Life Prediction Method for Farm Products Based on an Agricultural IoT, vol. 846, Trans Tech Publication, 2014, pp. 1830–1835.
[42] H.-T. Chang, N. Mishra, C.-C. Lin, IoT big-data centred knowledge granule analytic and cluster framework for BI applications: a case base analysis, PLoS One 10 (11) (2015) 1014–1980.
[43] H.M. Raafat, M.S. Hossain, E. Essa, S. Elmougy, A.S. Tolba, G. Muhammad, A. Ghoneim, Fog intelligence for real-time IoT sensor data analytics, IEEE Access 5 (2017) 24062–24069.
[44] M. Kontaki, A. Gounaris, A.N. Papadopoulos, K. Tsichlas, Y. Manolopoulos, Efficient and flexible algorithms for monitoring distance-based outliers over data streams, Inf. Syst. 55 (2016) 37–53.
[45] I. Vasconcelos, R.O. Vasconcelos, B. Olivieri, M. Roriz, M. Endler, M.C. Junior, Smartphone based outlier detection: a complex event processing approach for driving behavior detection, Journal of Internet Services and Applications 8 (1) (2017) 1–13.
[46] L. Lyu, J. Jin, S. Rajasegarar, X. He, M. Palaniswami, Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering, IEEE Internet Things J. 4 (5) (2017) 1174–1184.
[47] A. Daneshpazhouh, A. Sami, Entropy-based outlier detection using semi-supervised approach with few positive examples, Pattern Recogn. Lett. 49 (2014) 77–84.
[48] Xiaoling Wang, Z. Xu, C. Sha, M. Ester, A. Zhou, Semi-supervised learning from only positive and unlabeled data using entropy, in: Proceedings of International Conference on Web-Age Information Management, 2010, pp. 24–31.
[49] A. Delimargas, E. Skevakis, H. Halabian, I. Lambadaris, N. Seddigh, B. Nandy, R. Makkar, Proceedings of MILCOM 2015-IEEE Military Communications Conference, 2015, pp. 36–44.
[50] M. Rafferty, X. Liu, D.M. Laverty, S. McLoone, Real-time multiple event detection and classification using moving window PCA, IEEE Trans. Smart Grid 7 (5) (2016) 2537–2548.
[51] X. Kong, J. Chang, M. Niu, X. Huang, J. Wang, S.I. Chang, Research on real time feature extraction method for complex manufacturing big data, Int. J. Adv. Des. Manuf. Technol. 99 (5–8) (2018) 1101–1108.
[52] S. Cheng, Z. Cai, J. Li, H. Gao, Extracting kernel dataset from big sensory data in wireless sensor networks, IEEE Trans. Knowl. Data Eng. 29 (4) (2016) 813–827.
[53] K. Yan, D. Zhang, Feature selection and analysis on correlated gas sensor data with recursive feature elimination, Sensor. Actuator. B Chem. 212 (2015) 353–363.
[54] X. Li, S. Peng, J. Chen, B. Lü, H. Zhang, M. Lai, SVM–T-RFE: a novel gene selection algorithm for identifying metastasis-related genes in colorectal cancer using gene expression profiles, Biochem. Biophys. Res. Commun. 419 (2) (2012) 148–153.
[55] P. Szecowka, A. Szczurek, B. Licznerski, On reliability of neural network sensitivity analysis applied for sensor array optimization, Sensor. Actuator. B Chem. 157 (1) (2011) 298–303.
[56] D.R. Wijaya, R. Sarno, E. Zulaika, Sensor array optimization for mobile electronic nose: wavelet transforms and filters based feature selection approach, International Review on Computers and Software 11 (8) (2016) 659–671.
[57] Y.-H. Yuan, Q.-S. Sun, Fractional-order embedding multiset canonical correlations with applications to multi-feature fusion and recognition, Neurocomputing 122 (2013) 229–238.
[58] M. Haghighat, M. Abdel-Mottaleb, W. Alhalabi, Discriminant correlation analysis: real-time feature level fusion for multimodal biometric recognition, IEEE Trans. Inf. Forensics Secur. 11 (9) (2016) 1984–1996.

[59] Z. Wu, K. Mao, G.-W. Ng, Enhanced feature fusion through irrelevant redundancy elimination in intra-class and extra-class discriminative correlation analysis, Neurocomputing 335 (2019) 105–118.
[60] I. Mehmood, M. Sajjad, S. Baik, Mobile-cloud assisted video summarization framework for efficient management of remote sensing data generated by wireless capsule sensors, Sensors 14 (9) (2014) 17112–17145.
[61] J. Yu, M. Kim, H.-C. Bang, S.-H. Bae, S.-J. Kim, IoT as applications: cloud-based building management systems for the internet of things, Multimed. Tool. Appl. 75 (22) (2016) 14583–14596.
[62] M. Yuan, H. Sheng, Research on the fusion method of spatial data and multimedia information of multimedia sensor networks in cloud computing environment, Multimed. Tool. Appl. 76 (16) (2017) 17037–17054.
[63] C. Li, W. Wei, J. Li, W. Song, A cloud-based monitoring system via face recognition using Gabor and CS-LBP features, J. Supercomput. 73 (4) (2017) 1532–1546.
[64] R. Varatharajan, G. Manogaran, M. Priyan, A big data classification approach using LDA with an enhanced SVM method for ECG signals in cloud computing, Multimed. Tool. Appl. 77 (8) (2018) 10195–10215.
[65] Q. Zhang, C. Zhu, L.T. Yang, Z. Chen, L. Zhao, P. Li, An incremental CFS algorithm for clustering large data in industrial internet of things, IEEE Trans. Ind. Inf. 13 (3) (2007) 1193–1201.
[66] Aurora Gonzalez-Vidal, Payam Brnaghi, Antonio F. Skarmeta, BEATS: blocks of eigenvalues algorithm for time series segmentation, IEEE Trans. Knowl. Data Eng. 30 (11) (2018) 2051–2064.
[67] F. Bu, An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT, Future Generat. Comput. Syst. 88 (2018) 675–682.
[68] S. Misra, S. Chatterjee, Social choice considerations in cloud-assisted WBAN architecture for post-disaster healthcare: data aggregation and channelization, Inf. Sci. 284 (2014) 95–117.
[69] J.H. Abawajy, M.M. Hassan, Federated internet of things and cloud computing pervasive patient health monitoring system, IEEE Commun. Mag. 55 (1) (2017) 48–53.
[70] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, P. Hou, A new deep learning based food recognition system for dietary assessment on an edge computing service infrastructure, IEEE transactions on services computing 11 (2) (2018) 249–261.
[71] Akbar, A., Khan, A., Carrez, F., Moessner, K., Predictive analytics for complex IoT data streams, IEEE Internet Things J.. 4(5) (02017) 1571-1582.
[72] S. Lakshmanaprabu, K. Shankar, A. Khanna, D. Gupta, J.J. Rodrigues, P. R. Pinheiro, V.H.C. De Albuquerque, Effective features to classify big data using social internet of things, IEEE Access 6 (2018) 24196–24204.
[73] Wong Siaw Ling, Ooi Boon Yaik, Liew Soung Yue, A novel data reduction technique with fault-tolerance for internet-of-things, Association for Computing Machinery (ACM) 2 (1) (2017) 214–221.
[74] A. Ahmad, M. Khan, A. Paul, S. Din, M.M. Rathore, G. Jeon, G.S. Choi, Toward modeling and optimization of features selection in big data based social internet of things, Future Generat. Comput. Syst. 82 (2018) 715–726.
[75] C. Tong, X. Yin, S. Wang, Z. Zheng, A novel deep learning method for aircraft landing speed prediction based on cloud-based sensor data, Future Generat. Comput. Syst. 88 (2018) 552–558.
[76] M. Mohammadi, A. Al-Fuqaha, M. Guizani, J.-S. Oh, Semisupervised deep reinforcement learning in support of IoT and smart city services, IEEE Internet Things J. 5 (2) (2018) 624–635.
[77] H. Yan, J. Wan, C. Zhang, S. Tang, Q. Hua, Z. Wang, Industrial big data analytics for prediction of remaining useful life based on deep learning, IEEE Access 6 (2018) 17190–17197.
[78] H. Li, K. Ota, M. Dong, Learning IoT in edge: deep learning for the internet of things with edge computing, IEEE Network 32 (1) (2018) 96–101.
[79] C. Zhang, X. Pan, H. Li, A. Gardiner, I. Sargent, J. Hare, P.M. Atkinson, A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification, ISPRS J. Photogrammetry Remote Sens. 140 (2018) 133–144.
[80] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, P. Hou, A new deep learning based food recognition system for dietary assessment on an edge computing service infrastructure, IEEE transactions on services computing 11 (2) (2018) 249–261.
[81] P. Li, Z. Chen, L.T. Yang, Q. Zhang, M.J. Deen, Deep convolutional computation model for feature learning on big data in Internet of Things, IEEE Trans. Ind. Inf. 14 (2) (2018), 790798.
[82] S. Leroux, S. Bohez, E. De Coninck, T. Verbelen, B. Vankeirsbilck, P. Simoens, B. Dhoedt, The cascading neural network: building the Internet of Smart Things, Knowl. Inf. Syst. 52 (3) (2017) 791–814.
[83] J. Zhang, H. Huang, X. Wang, Resource provision algorithms in cloud computing: a survey, J. Netw. Comput. Appl. 64 (2016) 23–42.
[84] A. Singh, D. Juneja, M. Malhotra, Autonomous agent based load balancing algorithm in cloud computing, Procedia Comput. Sci. 45 (2015) 832–841.
[85] D.C. Devi, V.R. Uthariaraj, Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks, Sci. World J. 3 (9) (2016) 111–121.
[86] W. Jing, Q. Miao, G. Chen, An open scheduling framework for QoS resource managemnt in the internet of things, KSII Transactions on Internet and Information Systems 12 (9) (2018) 4103–4121.
[87] J. Li, Z. Lu, W. Zhang, J. Wu, H. Qiang, B. Li, P.C. Hung, SERAC3: smart and economical resource allocation for big data clusters in community clouds, Future Generat. Comput. Syst. 85 (2018) 210–221.
[88] G.J.L. Paulraj, S.A.J. Francis, J.D. Peter, I.J. Jebadurai, Resource-aware virtual machine migration in IoT cloud, Future Generat. Comput. Syst. 85 (2018) 173–183.
[89] G. Shidik, A. Azhari, K. Mustofa, Improvement of energy efficiency at cloud data center based on fuzzy markov normal algorithm vm selection in dynamic vm consolidation, International Review on Computers and Software (IRECOS). 11 (6) (2016) 511–520.
[90] A. Beloglazovy, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurrency Comput. Pract. Ex. 24 (13) (2011) 1–24.
[91] H. Li, W. Li, H. Wang, J. Wang, An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud, Future Generat. Comput. Syst. 84 (2018) 98–107.
[92] M. Elhoseny, A. Abdelaziz, A.S. Salama, A.M. Riad, K. Muhammad, A.K. Sangaiah, A hybrid model of internet of things and cloud computing to manage big data in health services applications, Future Generat. Comput. Syst. 86 (2018) 1383–1394.
[93] H.S. Narman, M.S. Hossain, M. Atiquzzaman, H. Shen, Scheduling internet of things applications in cloud computing, Annals of Telecommunications 72 (1–2) (2017) 79–93.
[94] C.-J. Huang, C.-T. Guan, H.-M. Chen, Y.-W. Wang, S.-C. Chang, C.-Y. Li, C.-H. Weng, An adaptive resource management scheme in cloud computing, Eng. Appl. Artif. Intell. 26 (1) (2013) 382–389.
[95] R. Jeyarani, N. Nagaveni, R.V. Ram, Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence, Future Generat. Comput. Syst. 28 (5) (2012) 811–821.
[96] N.T. Hieu, M. Di Francesco, A.Y. Jääski, A virtual machine placement algorithm for balanced resource utilization in cloud data centers, in: Paper Presented at the 2014 IEEE 7th International Conference on Cloud Computing, 2014.
[97] M.S. Mekala, P. Viswanathan, Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for IoT, Comput. Electr. Eng. 73 (2019) 227–244.
[98] H. Hallawi, J. Mehnen, H. He, Multi-Capacity Combinatorial Ordering GA in Application to Cloud resources allocation and efficient virtual machines consolidation, Future Generat. Comput. Syst. 69 (2017) 1–10.
[99] I. Mohiuddin, A. Almogren, Workload aware VM consolidation method in edge/cloud computing for IoT applications, J. Parallel Distr. Comput. 123 (2019) 204–214.
[100] M.M. Abed, M.F. Younis, Developing load balancing for IoT-cloud computing based on advanced firefly and weighted round robin algorithms, Baghdad Science Journal 16 (1) (2019) 130–139.
[101] Y.-H. Chen, C.-Y. Chen, Service oriented cloud VM placement strategy for Internet of Things, IEEE Access 5 (2017) 25396–25407.
[102] S. Basu, M. Karuppiah, K. Selvakumar, K.-C. Li, S. Islam, H. K, M.M. Hassan, Md Bhuiyan, A. Z, An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment, Future Generat. Comput. Syst. 88 (2018) (2018) 254–261.
[103] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: Proceedings of 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST), 2010, pp. 21–27.
[104] H. Yaish, M. Goyal, G. Feuerlicht, Multi-tenant elastic extension tables data management, Procedia Comput. Sci. 29 (2014) (2014) 2168–2181.
[105] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L.T. Yang, Data mining for internet of things: a survey, IEEE communications surveys & tutorials 16 (1) (2014) 77–97.
[106] F. Samie, L. Bauer, J. Henkel, From cloud down to things: an overview of machine learning in internet of things, IEEE Internet Things J. 12 (2019).
[107] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Network. 61 (2015) 85–117.
[108] A. Ali, J. Qadir, R. ur Rasool, A. Sathiaseelan, A. Zwitter, J. Crowcroft, Big data for development: applications and techniques, Big Data Analytics 1 (1) (2016) 2–24.
[109] G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, C. I. Sánchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88.
[110] M.H. Rehman, V. Chang, A. Batool, T.Y. Wah, Big data reduction framework for value creation in sustainable enterprises, Int. J. Inf. Manag. 36 (6) (2016) 917–928.
[111] S. Kraijak, P. Tuwanut, A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends, in: Proceedings IEEE 16th International Conference on Communication Technology, ICCT), 2015, pp. 23–30.
[112] D. Soni, A. Makwana, A survey on MQTT: a protocol of internet of things (IoT), in: Proceedings of the International Conference on Telecommunication, Power Analysis and Computing Techniques, ICTPACT), 2017, pp. 1–6.
[113] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE communications surveys and tutorials 17 (4) (2015) 2347–2376.
[114] C. Gomez, J. Oller, J. Paradells, Overview and evaluation of bluetooth low energy: an emerging low-power wireless technology, Sensors 12 (9) (2012) 11734–11753.
[115] J.W. Hui, D.E. Culler, Extending IP to low-power, wireless personal area networks, IEEE Internet Computing (4) (2008) 37–45.
[116] J. Nieminen, B. Patil, T. Savolainen, M. Isomaki, Z. Shelby, C. Gomez, Transmission of IPV6 Packets over Bluetooth Low Energy Draft-Ietf-6lowpan-Btle-

12, 2012. https://datatracker.ietf.org/doc/html/draftietf-6lowpan-btle-06. (Accessed 17 June 2019).

[117] E. Dahlman, S. Parkvall, J. Skold, 4G: LTE/LTE-advanced for Mobile Broadband, second ed.), second ed., Academic press United Kingdom and United State of America, 2013.

[118] F. Schaich, B. Sayrac, M. Schubert, H. Lin, K. Pedersen, M. Shaat, A. Georgakopoulos, FANTASTIC-5G: 5G-PPP Project on 5G air interface below 6 GHz, in: Proceedings at the European Conference on Network and Communications, 2015, pp. 1–7.

[119] OASIS, Advanced Message Queuing Protocol (AMQP) Cliams-Based Security Version, 1.0, 2017. https://www.oasis-open.org/committees/.../amqp-cbs-v1.0-wd04.doc. (Accessed 12 May 2019).

[120] F.T. Johnsen, T.H. Bloebaum, M. Avlesen, S. Spjelkavik, B. Vik, Evaluation of transport protocols for web services, in: Proceedings of 2013 Military Communications and Information Systems Conference, 2013, pp. 54–62.

[121] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, J. Alonso-Zarate, A survey on application layer protocols for the internet of things, Transaction on IoT and Cloud computing 3 (1) (2015) 11–17.

[122] S. Song, B. Issac, Analysis of wifi and wimax and wireless network coexistence, Journal of Computer Networks and Communications (IJCNC). 6 (6) (2014) 63–78.

[123] R. Hai, S. Geisler, C. Quix, Constance: an intelligent data lake system, in: Proceedings of the International Conference on Management of Data, 2016, pp. 11–18.

[124] A. Meddeb, Internet of things standards: who stands out from the crowd? IEEE Commun. Mag. 54 (7) (2016) 40–47.

[125] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: a survey, IEEE communications surveys & tutorials 16 (1) (2014) 414–454.