Internet of Things 19 (2022) 100516

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Review article

Lossy Data Compression for IoT Sensors: A Review

Juan David Arias Correa [a,*], Alex Sandro Roschildt Pinto [a,b], Carlos Montez [a]

[a] Federal University of Santa Catarina, Dept. of Automation and Systems, Campus Universitário Trindade, Florianópolis, 88040-900, Santa Catarina, Brazil

[b] Federal University of Santa Catarina, Dept. of Informatics and Statistic, Campus Universitário Trindade, Florianópolis, 88040-370, Santa Catarina, Brazil

ARTICLE INFO

Keywords:
Data compression
IoT
Internet of things
Lossy compression
Sensors

ABSTRACT

Internet of Things (IoT) can be considered a suitable platform for industrial applications, enabling large systems that connect a huge number of intelligent sensors and subsequent data collection for analytical applications. This factor is responsible for the substantial increase in the current volume of data generated by IoT devices. The large volume of data generated by IoT sensors can lead to unusual demands on cloud storage and data transmission bandwidths. A suitable approach to address these issues is through data compression approaches. This article presents a systematic review of the literature on lossy data compression algorithms that allows the systems to reduce the data detected by IoT devices. Lossy algorithms have a good compression ratio, preserving data quality and minimizing compression errors. A taxonomy was proposed from the review results, and the main works were classified, analyzed, and discussed.

1. Introduction

Internet of Things (IoT) concretizes the vision of connecting real-world objects with digital ones, making the separation between physical and digital worlds transparent. "Things" are intelligent objects that can be embedded with software, services and technologies. These objects often comprise small devices with computing and storage capacity, which allow them to perform tasks and communicate with other objects and systems without direct human intervention. By allowing objects to connect their services, resources and intelligence to each other, IoT supports the creation of a new expanded Internet [1–4].

IoT usage has been a heavy increase in the last ten years. Since 2011, the number of interconnected objects has exceeded the human population [5], which has driven the emergence of distributed applications on the Internet such as intelligent transportation systems, smart home, smart healthcare, and disaster monitoring, among others [6]. It is estimated that a few tens of billions of IoT nodes are going to be connected by the end of 2021 [7] and 75 billion by 2025 [8].

Recent trends of applying IoT and cyber–physical system (CPS) concepts in industrial application scenarios have led to the development of the industry 4.0 concept [9], which, in turn, is one of the factors responsible for the substantial current increase in the volume of generated data by IoT devices. In the context of Industry 4.0, IoT may be considered a suitable platform that will allow the implementation of large systems that connect a huge number of smart sensors and subsequent data collection for analytic applications [9].

However, the large volume of data generated by IoT devices entails notable storage and transmission costs [10,11]. Communication systems constitute essential infrastructure for IoT objects, and even with the emergence of low-power networks [12], these systems are responsible for the significant energy consumption of devices. Due to the power limitations of some IoT systems, this

* Corresponding author.
E-mail address: juan.arias@posgrad.ufsc.br (J.D.A. Correa).

https://doi.org/10.1016/j.iot.2022.100516
Received 10 August 2021; Received in revised form 14 February 2022; Accepted 3 March 2022
Available online 30 March 2022
2542-6605/

consumption becomes a problem that must be faced [13–15]. Processing a large dataset requires even more computational resources and execution time. To make this problem worse, the large volume of sensed data that needs to be stored in fogs and clouds incurs a high monetary cost, as these systems tend to charge proportional values for the stored data [16,17].

These issues can be addressed from hardware or communication point of views, such as by selecting a suitable communication protocol or increasing the network bandwidth or the fog and cloud storage capacity. However, the most common approach to deal with these IoT problems is through a semantic-oriented point of view [18], which addresses topics related to the best ways to represent, store and organize information. In this context, data compression (DC) approaches play a key role in meeting communication infrastructure and scalable storage requirements, becoming fundamental approaches to managing a large amount of generated information.

Modern DC techniques have been developed since the first digital computers in the 1940s and have been addressed in different research areas such as information theory, signal processing, multimedia, telecommunications and computer security. There are numerous DC definitions, which depend on the target application. In the context of data generated by IoT sensors, which is the topic of interest in this article, DC can be defined as mechanisms or techniques designed for reducing the number of samples necessary to represent a signal for saving storage space or transmission time [19].

Over the years, diverse taxonomies for DC techniques have been proposed. Techniques may be categorized as, for example, static or dynamic (concerning whether the codewords change over time), symmetric or asymmetric (referring to the relationship between compression and decompression execution times), semantic-dependent or entropy-based (referring to whether data semantics is exploited or not for better compression), variable or fixed-rate (referring to data encoding rate), and lossless or lossy (regarding the decompression results) [20–24].

In this article, we assume that classification into lossless or lossy compression should be the starting point for the study of DC techniques in the context of data generated by IoT sensors. Lossless algorithms are those capable of reconstructing a signal exactly in the form of its original representation. On the other hand, lossy algorithms allow reconstructing only an approximation of the original signal [17], in order to represent the signal with fewer samples than lossless algorithms. In lossy approaches, sensor measurements are commonly represented by eliminating redundant samples or representing them in a compressed form, with the premise that the redundant samples would not provide additional information for the application [6].

The focus on lossless or lossy classification is because sampled IoT sensor data is often used to represent variable signals [25], and it is crucial to address DC techniques, analyzing the resulting signal after the compression and decompression steps. The rationale is that after the decompression process, the resulting signal would later be used for analytics (e.g., in machine learning approaches) or decision-making (e.g., to trigger alarms or control actuators) in various systems. This article presents a Systematic Literature Review (SLR) of lossy algorithms. There are some reasons to address lossy techniques. One of them is that they tend to get similar results to lossless DC, with computationally less complex algorithms that offer a better compression ratio. Lossy algorithms allow obtaining similar results using fewer samples than lossless. These reasons make lossy DC techniques excellent alternatives for IoT devices, improving the management of energy, storage, and communication resources.

In the SLR selection performed on this article, after an initial gathering of 314 research articles, a methodical selection resulted in 25 documents, through the use of selection and exclusion criteria. The documents were selected mainly considering whether their proposals are for sensors and IoT devices with processing, storage, and energy restrictions. The SLR results were classified by a proposed taxonomy, with a summary of each proposal, discussing the type of input and output of the DC mechanisms, the parameters needed, and their particularities.

The remainder of this article is organized as follows. Section 2 presents the basics of data compression with a focus on lossy DC techniques. Section 3 briefly discusses other reviews of the literature on lossy compression. Section 4 details the SLR protocol, and proposes a taxonomy created from the SLR. Section 5 presents the SLR results. Finally, Section 6 summarizes the work and concludes the article.

2. Data compression

Data compression can be roughly defined as the process of representing some content, reducing the amount of data needed to represent it. In IoT context, DC mechanisms reduce the size of the information that must be transmitted or stored, decreasing the consumption of resources such as storage systems, communication bandwidth, or energy consumption [16,17,23,26].

Morse code in the 19th century is cited as one of the first examples of DC. However, DC theory received a strong boost in the 1940s with Claude Shannon's entropy theory. Currently, compression techniques are fundamental in applications such as satellite imagery, geographic information systems (GIS), graphics, Wireless Sensor Networks (WSN), IoT and others [27]. DC is commonly used in audio and video compression, image processing, data transfer, digital telecommunication network and others [23].

DC basically has two process, the compression process that transform the original data in a compressed way, and the decompression process, which returns the original data or an approximation thereof. These DC processes cannot be separately analyzed as they are interdependent but taking into account the decompression results, the DC techniques can be broadly classified into lossless and lossy.

Lossless techniques transform the data in a compressed form and allow the reconstruction of original data exactly from the compressed one. These techniques are also called reversible compression. They are normally used for data that cannot be modified and the decompress process needs to return the original data, such as database records, spreadsheets, and document files, among others. Adaptive Huffman coding, Arithmetic coding and LZSS are examples of lossless approaches [23,26].

Table 1
PICOC criteria used for this study.

Population	Sensed data
Intervention	Lossy Data Compression algorithms for IoT devices
Outcomes	Better use of resources and cost reduction
Context	IoT devices connecting with a server (local host, fog or cloud)

Lossy techniques can only reconstruct an approximation of the original data. These techniques are also called irreversible compression because it is impossible to recover the original data. Lossy approaches tend to have less computational complexity and higher compression results than lossless. They are commonly used in applications that allow eliminating redundant information as sensor data streams (from periodic sensor sampling), graphics, images and sounds [17]. Swinging Door Trending [28], linear approximation, DWT, MP3 and JPEG codification are examples of lossy approaches.

The massive increase in the number of sensors deployed in recent years due to the popularization of IoT, the digitization of information due to Industry 4.0, among others, have caused an increase in the amount of data generated and the consequent need for more communication bandwidth and storage capacity. To make matters worse, currently, the rate of growth of data is much higher than the rate of growth of storage and transmission technologies [27]. Lossy DC techniques are taking their place as a possible mechanism to deal with this kind of challenge.

3. Related work

In the literature it is possible to find some reviews about DC algorithms that can be used in sensor measurement scenarios. However, most of them are not exclusively for lossy DC approaches. This section briefly discusses them.

Bose et al. [29] present an analysis of lossy compression algorithms and classify them based on the signal characteristics of sensor data. The algorithms are categorized into time and transform domains.

Srisooksai et al. [22] perform a comprehensive review of existing DC approaches for WSNs. The algorithms are classified between local and distributed approaches. The document has a lot of important information and explains properly the review process done; however, the article was published in 2012, being all discussed articles prior to 2010.

Tuama et al. [30] present an overview of the recent advances of DC in WSN. The algorithms are separated into distributed and local approaches, which in turn are classified in lossy and lossless. The review presents 16 articles but only two are about lossy DC techniques.

Uthayakumar et al. [31] present a survey of DC techniques from data quality, coding scheme, data type, and applications perspectives. The article considers WSN applications, in which only one lossy DC technique is discussed.

Reviews about lossless compression techniques are performed by both Kotha et al. [23] and Dolfus and Braun [26]. This second review focused only on WSN approaches.

A survey on data compression in WSN published in 2005 is presented in Kimura and Latifi [32]. The authors discuss five types of DC schemes (coding by ordering, pipelined in-network compression, JPEG 2000, low-complexity video compression, and distributed compression).

To better address the above-discussed issues, this paper presents a systematic review of lossy DC techniques. The main contribution of our study is to present an SLR about lossy DC techniques aimed at IoT applications, mainly in the industrial context, in which there is a high cost in transmitting and storing data streams generated by periodic sampling of sensors.

4. SLR protocol

Systematic literature review is a type of literature review that allows the discovery, examination and categorization of the existing literature in an organized and methodical way, providing a basis for experts to suggest future research. In this article, a SLR was carried out in January 2022 based on the guidelines proposed in [33]. In this sense, the starting point of an SLR is through a research question (RQ). The basic RQ used was: **"What are the lossy algorithms for data compression for sensed data in IoT devices?"**.

To focus on the target application area, this RQ was subdivided into more specific questions: **"In the industrial and CPS contexts, are there relevant works on lossy data compression for IoT sensors?"** and **"Are there data compression techniques that allow cost savings in storing data in fogs and clouds?"**.

It is necessary to convert these RQs into concrete search keywords, and the first step is by creating PICOC criteria (Population, Intervention, Comparison, Outcomes, Context) [33].

Table 1 shows the criteria used in this study.

PICOC criteria are fundamental because they delimit the context of the study. In short, the objective was to find articles on lossy DC algorithms for sensor data in IoT devices with power and processing constraints in order to improve the use of resources (network bandwidth, memory space on fog and servers) and reduce economic costs.

In a second step, these criteria were refined to find keywords to use in the bibliographic search.

Table 2 presents the resulting keywords.

The remainder of the SLR tasks consists of four stages [33]. The first stage uses the PICOC keywords as search strings to find articles that meet the basic conditions. The second stage applies selection and exclusion criteria in the abstract of the documents. The third stage applies the same criteria to the documents' full texts. The fourth stage consists of data extraction and synthesis to be used in this review. Fig. 1 presents a general overview of the SLR protocol and the number of documents involved in each stage. These stages and the results obtained are presented in the following sections.

Table 2
PICOC keywords.

Population	Sensed data, sensing data, sensor data, IoT data
Intervention	Data compression, data compressing, compress data
Outcomes	Reduction, saving, cost, reducing
Context	IoT, IoE, SCADA, Industry 4.0, CPS, Smart Industry, WSN

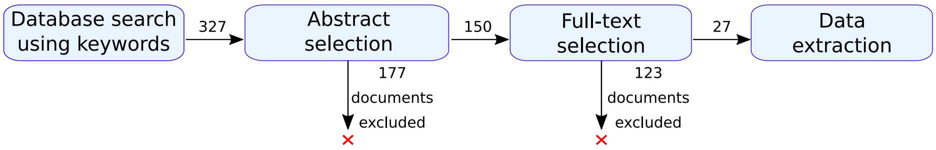

Fig. 1. Overview of the documents selection process.

4.1. Database search using keywords

The search using the PICOC keywords was carried out in the Scopus database. This database indexes journal and conference articles from various publishers. The publishers covered by this database comprise essential areas for this SLR such as Engineering (e.g., IEEE Xplore, ScienceDirect) and Computer Science (e.g., ACM, Elsevier, Emerald, IOS Press, Springer, Taylor & Francis and Wiley).

Scopus database was also selected because of its query interface, which is very flexible and allows for the expression of complex queries. The W/n operator, for example, allows defining the maximum number of words (represented by n) within two keywords. This operator can be used to expand search results. Listing 1 presents the search string that was derived from the keywords in Table 2. This search string resulted in 327 documents.

```
TITLE-ABS-KEY
(((data W/2 (sensed OR sensing OR sensor OR IoT )) W/5
  (compression OR compressing OR compress)
 )AND
 (iot OR "internet of things" OR ioe OR "internet of everything" OR scada
  OR "industry 4.0" OR cps OR "cyber-physical system"
  OR "smart industry" OR wsn OR "Wireless sensor network"
 )
) AND DOCTYPE(ar OR cp OR ip) AND LANGUAGE(english)
  AND SUBJAREA(engi OR comp)
```

Listing 1: Scopus search string.

4.2. Selection criteria

The second and third stages consist of initially submitting the abstracts and later the full texts of the articles to criteria that exclude those that do not meet minimum requirements. These selection criteria are divided into inclusion criteria (which must be met) and exclusion criteria (which must not be met). In this SLR, selected articles must meet all inclusion criteria and cannot match any of the exclusion criteria.

The inclusion criteria were created to be applied in the initial stage and consist of simple rules. In short, the document must propose algorithms or methods for data compression; and the algorithm/method must be applied to a stream of sensor data.

Ten exclusion criteria were used. Briefly, the first rules used were to select only lossy compression algorithms applied to digital signals, in which the compression is performed inside IoT devices. In addition, only studies in English published in peer reviews and related to engineering or computer science must be selected. Other exclusion criteria concern the article's content, for example, if the entire text does not provide sufficient detail about the method used (e.g., incomplete information. conclusion without theoretical or empirical support), or if the document has an extended version that presents the method in more detail.

Table 3 presents a summary of the excluded articles due to inclusion and exclusion criteria. It is possible to observe that the inclusion criteria rejected more articles in the abstract analysis phase (e.g., articles out of scope), and the exclusion criteria served to eliminate more articles in the full-text analysis phase (e.g., articles without technical results)

In total, the selection criteria rejected 92% of the articles (300 out of 327). Only 27 works remained to be analyzed in this SLR. In the selected documents, 12 publications were in journals and 15 in conferences.

Table 4 shows the selected scientific journals by their publisher.

Finally, the selected 27 articles were submitted to the data extraction and synthesis phase.

Table 3
Excluded documents due to inclusion and exclusion criteria.

	Inclusion criteria	Exclusion criteria	Total excluded
Abstract	126 (71%)	51 (29%)	**177**
Full-text	45 (37%)	78 (63%)	**123**

Table 4
Number of journals per publisher.

IEEE	SAGE	Springer	ACM	MDPI	Inderscience	ISA	Elsevier	ASCE
3	2	1	1	1	1	1	1	1

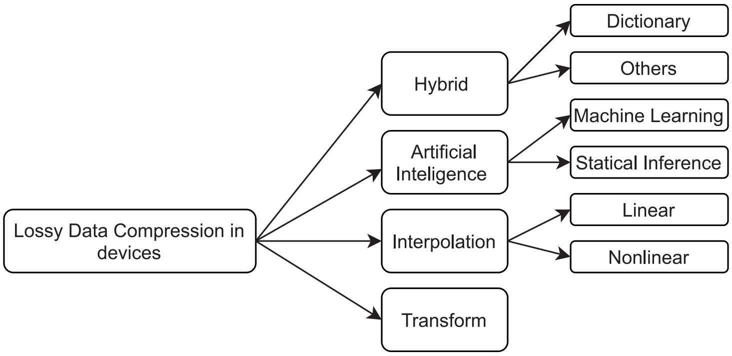

Fig. 2. Taxonomy of lossy algorithms for IoT sensors.

4.3. Data extraction and synthesis

The data extraction process consists of writing a summary for each document and answering a set of questions related to questions such as "what are the input, output, configuration, features and limitations of the algorithm? A synthesis of the data extraction results is then performed. The main output of this data synthesis is to identify the sensor compression algorithms that have been proposed and classify them. A taxonomy was created based on the techniques surveyed by the SLR and it is illustrated in Fig. 2.

In short, **Hybrid** (H) methods use lossy and lossless algorithms working together. They are divided into **Dictionary** (D) and **Others** (O) approaches, where Dictionary methods focus on methods that use a data dictionary for improving the compression rate. **Artificial Intelligence** (AI) proposals use methods of **Machine Learning** (ML) or **Statistical Inference** (SI). In **Interpolation** (I) methods, sensor signals are represented as the value of a function, and new data points are created using a set of known data points. These methods are divided into two subcategories, **Linear** (LI) and **Nonlinear** (NLI) methods. Finally, **Transform** (TR) methods compress data changing the domain of the signal (e.g., time to frequency domain).

The proposals covered by this taxonomy are discussed in detail in the next section.

5. Taxonomy results

This section synthesizes the data extracted from the selected documents in the form of taxonomy and discusses the articles by grouping them. Fig. 3 presents the proposals regarding their publication year and their classification in the taxonomy presented in Fig. 2. The oldest document was published in 2005. Half of the proposals were published between 2015 and 2021. The classification results show that twelve proposals employ interpolation approaches, ten are hybrid approaches, five use AI mechanisms, and three perform transform processes. AI approaches have increased in the last five years, which can be attributed to the growing interest in using AI in IoT areas to implement smart objects. However, most AI proposals are still proposed to be implemented in clouds or gateways. For this reason, this recent growth still does not appear in the methods analyzed in this SLR.

Table 5 summarizes the basic information of the selected documents. It includes the proposed technique of each selected work and its classification in the taxonomy. As two of the selected articles present more than one approach, the number of proposals in the table is higher than the number of articles (30 proposals).

5.1. Hybrid

Combining lossy and lossless algorithms is a recurrent approach. These approaches have good results for compression ratio (CR) and compression error (CE), two commonly used performance metrics. They can take into account the system situation to improve the compression process or to recognize system needs. Hybrid approaches tend to have more steps and more computational demands than other types of DC proposals.

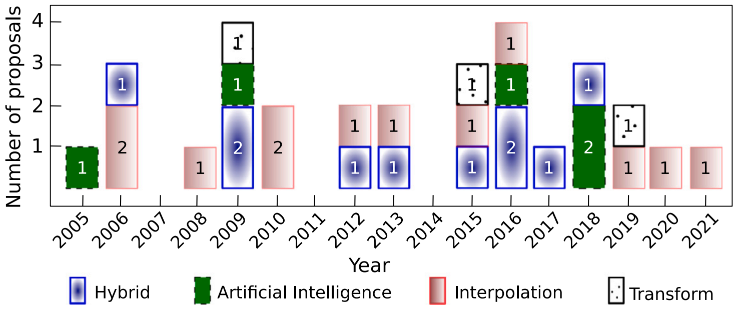

Fig. 3. Proposals per year.

Table 5
General information of selected documents (sorted by year).

Document	Proposed technique	Type
Li and Li [34]	Precision Based Sampling and Transmit Algorithm (PBSA)	AI/SI
Li et al. [35]	Linear approximation	I/LI
Li et al. [35]	Bezier curve approximation	I/NL
Zhang and Li [36]	Lifting Scheme Wavelet Transform (LSWT)	H/O
Pham et al. [37]	Top-Down Piecewise Linear Approximation with Minimum number of Line Segments	I/LI
Capo-Chichi et al. [38]	K-precision Run Length (K-RLE)	H/O
Liu and Yu [39]	Wavelet Transform with Arithmetic encoding	H/O
Liu and Yu [39]	Wavelet packet analysis with the Fisher-information	TR
Liu and Yu [39]	Monotone increase compression	AI/SI
Chen et al. [40]	Nonthreshold-based node level Algorithm	I/LI
Kasirajan et al. [41]	Non-linear Adaptive Pulse Coded Modulation-Based Compression (NADPCMC)	I/NL
Chen et al. [42]	Dynamic Bounded-Error DC and Aggregation (D-BEDCA)	H/D
Kui Zhao et al. [43]	Distributed Swinging Door Trending (DSDT)	I/LI
Mohamed et al. [44]	Adaptive DC based on prediction techniques	H/O
Zhang et al. [45]	Self-Adaptive Regression-based Multivariate Data Wavelet Compression Scheme with Error Bound (AR-MWCEB)	I/LI
Sharma [46]	Lightweight Temporal Compression (LTC)	I/LI
Alsalaet and Ali [47]	Modified Discrete Cosine Transform with Embedded Harmonic Components Coding (MDCT-EHCC)	H/O
Bashlovkina et al. [48]	Fuzzy Compression Adaptative Transform (FuzzyCat)	TR
Li and Liang [49]	Two-modal Generalized Predictive Coding	H/D
Abu Alsheikh et al. [50]	Error Bound using Compression Neural Networks	AI/ML
Kolo et al. [51]	Efficient and Robust Adaptive DC Scheme (ADCS)	H/D
Wang et al. [52]	Tree-structure linear approximation scheme	I/LI
Giorgi [53]	Zero-latency predictive filter base on DPCM	H/O
Liu et al. [54]	Stacked RBM Auto-Encoder (Stacked RBM-AE)	AI/ML
Ramijak et al. [55]	Approximate Vector Stream Compression (AVSC)	H/D
Harb et al. [56]	Two-level data reduction using Pearson coefficients and EKmeans or TopK	AI/SI
Azar et al. [57]	Fast error-bounded lossy compressor	I/NLI
Chen et al. [58]	Hierarchical Adaptive Spatio-Temporal DC (HASDC)	TR
Mendes et al. [59]	Adaptive Data Compression Mechanism	I/NL
Klus et al. [60]	Direct Lightweight Temporal Compression (DLTC)	I/LI

Table 6 presents a general review of the proposal's characteristics. Input, output, parameters and features to improve compression results are presented.

Table 6
Hybrid approaches information.

Document	Input	Output	Parameters	Features
Ramijak et al. [55]	Time series	Codeword	Number of bins for discretization, Max Euclidean distance, Pre-defined seq. length	Error bound using SQS
Chen et al. [42]	Sample	Codeword	Max Error	Dynamic error bounded by temporal, spatial or data correlation
Kolo et al. [51]	Array	Table ID+ codeword	SME, Static Huffman tables	Static Huffman tables, Adaptive compression
Mohamed et al. [44]	Sample	Codeword	Error range	Adaptive error bound, Event detection, Energy and data quality aware
Giorgi [53]	Units Sample	Codeword	Tolerance threshold	
Alsalaet and Ali [47]	Array	Codeword	Array size	Embedded coding scheme
Capo-Chichi et al. [38]	Sample	Value count	Tolerance threshold	
Li and Liang [49]	Sample	Codeword	Compression radius Residue error bound	
Zhang and Li [36]	Vector	Entropy coding	Quantization size	
Liu and Yu [39]	Vector	Codeword	Decomposition level Number of samples	

5.1.1. Dictionary

Many DC algorithms use dictionaries to represent the compressed result. The storage of a dictionary on devices increases storage resource needs. Some dictionaries work as lossless algorithms (e.g., Huffman coding), but they can be used as lossy algorithms with an additional process. These lossy proposals decrease the dictionary size and the necessary bits to represent the sensor data. They add some errors to the system but improve the compression rate.

Ramijak et al. [55] propose the Approximate Vector Stream Compression (AVSC) that has an uncorrelated (AVSC-NC) and a Correlate (AVSC) version. AVCSC uses Summarizing event seQuenceS (SQS) before executing a dictionary algorithm. AVSC ensures that the sequence of events can be derived (or inferred) from the compressed data.

Chen et al. [42] propose the Dynamic Bounded-Error Data Compression and Aggregation (D-BEDCA). It is an improvement of BEDCA that was proposed in [61]. The approach has two phases. First is calculated the Coefficient Variation (CV). The error bound is assigned based on CV for each sensor node. The second is to compress under this error bound considering the temporal (previous sample), spatial (neighboring sample), or data correlation (the last codeword). D-BEDCA uses a dictionary algorithm proposed in [62] for encoding. The temporal correlation process is performed at the sensor node and the data and spatial correlation in an external node (sink node). The bounded-error mechanisms do not consider the trend of the data, only the difference with the last sample.

Kolo et al. [51] propose an Efficient and Robust Adaptive Data Compression Scheme (ADCS) that changes between lossless and lossy depending on the application. The main component of ADCS is the Adaptive Entropy Encoder (AEE) that is a lossless approach. AEE uses two static Huffman tables that were designed to handle different levels of correlation in the source data. These tables are made out of the device for reducing the computational complexity. The brute-force and the decision region approach are used to encode blocks of data efficiently at a time using multiple tables adaptively. When necessary, Sensor Manufacture Error (SME) is used as an error bound for filtering the signal. The proposal increases the computational complexity by deciding which table to use. An extra bit was utilized to identify which table was used to encode.

Li and Liang [49] propose a Generalized Predictive Coding (GPC) framework and a lossless algorithm called Sequential Lossless Entropy Compression (S-LEC). GPC and S-LEC form a powerful unified lossless and lossy compression solution. The lossy component uses an error bound before the predictive coding. The alphabet is done based on residue groups. The error is bounded considering the residue generated in each sensed sample. When the sample does not exceed this margin, the residue is considered equal to zero.

5.1.2. Others works

Mohamed et al. [44] proposes a framework that decides between a lossless or lossy compression method according to the available energy and the criticality of the data. Markov Decision Process (MDP) is used to optimize the data quality and energy

Table 7
AI information.

Document	Input	Output	Parameters	Features
Liu et al. [54]	Array	Compressed array	Training data Number of iterations Learning rate	Energy optimization, Pre-training process
Abu Alsheikh et al. [50]	Array	Compressed array	Training data Tolerance threshold	Outlier detector Offline compression dictionary
Li and Li [34]	Sample Time series	Sample Group+interval	Tolerance error Sampling period	Sampling frequency
Harb et al. [56]	Array	Representative array	Period size Pearson's threshold	Elimination of redundant data generated by neighbors
Liu and Yu [39]	Array	Component set	Compression rate	

savings. Event detection is used to assess whether the data is important (lossless sending) or not (compressed). An entropy encoder is used for lossless DC. The proposal uses an error radius to define how much error is acceptable. It works as an error bounder that decides when a sample will be encoded and sent.

Giorgi [53] proposes a zero-latency compression algorithm that combines lossy and lossless techniques. Lossy compression is performed first and is a zero-latency predictive filter based on differential pulse code modulation (DPCM). It uses a tolerance threshold to decide when necessary to pass the sample to the lossless technique. The lossless technique uses a modified Exponential Golomb code with a fixed-length prefix.

Alsalaet and Ali [47] present a DC scheme based on Modified Discrete Cosine Transform (MDCT) followed by Embedded Harmonic Components Coding (EHCC) called MDCT-EHCC. The proposal uses MDCT as the lossy algorithm. An N-sample signal is represented by MDCT coefficients. Additionally, EHCC is used to improve the CR. EHCC provides embedded coding that allows to transmit the coefficients progressively according with their significance. The idea behind this scheme is explore harmonic redundancy.

The K-precision Run Length Encoding (K-RLE) is proposed by Capo-Chichi et al. [38]. RLE is a lossless algorithm. K-precision adds an error-bound system using a tolerance threshold (K). The proposal is simple and suitable for tiny devices with low computational resources. The obtained CR is low and depends on the signal stability.

Zhang and Li [36] present a vibration sensor data compression technique based on the lifting Scheme Wavelet Transform (LSWT), which takes into account the time–frequency characteristics of the sensor data. The proposal consists of a process with three steps: decorrelation (Lifting Scheme Haar Wavelet Transform), quantization and entropy coding. This proposal focuses exclusively on vibration sensors and their behavior.

Liu and Yu [39] propose the use of wavelet Transform with arithmetic encoding. This proposal focuses on WSN with a constant communication flow within sensors on Intelligent Transportation Systems (ITS). The proposal does not offer good results in terms of EC.

5.2. Artificial intelligence

AI allows addressing challenges of different computer science and engineering areas.

Table 7 presents a summary of the proposals that use AI approaches for data compression.

5.2.1. Machine learning

Machine learning is an important tool that can address many issues. It is a potential mechanism to create or improve compression methods.

Liu et al. [54] propose the use of Restricted Boltzmann Machines (RBMs) combined with non-linear learning methods from deep learning theory to create a new DC method. This proposal is called Stacked RBM Auto-encoder (Stacked RBM-AE). Stacked RBM-AE is highly complex to be used in IoT devices. It can be used on devices with good computational resources and is only recommended for applications with low error tolerance.

Abu Alsheikh et al. [50] propose a data compression algorithm with error bound using Compressing Neural Networks. The algorithm was created to have low computational complexity and uses only linear and sigmoidal operations. The algorithm uses the temporal–spatial correlation and can be used for both temporal and spatial compression. The algorithm works using a learned decompression dictionary that is necessary to recover the data. The neural autoencoders (AEs) is used, which is a three-layer neural network. The computational complexity is $O(L \times K)$, where L is the input data size and K is the compressed data size.

Table 8
Interpolation information.

Document	Input	Output	Parameters	Features
Kui Zhao et al. [43]	Time series	Representative time point	Error threshold	Distributed compression
Pham et al. [37]	Time series	Representative time series	Error threshold	Reduction of computational complexity
Wang et al. [52]	Array	Representative array	Window size, Distortion target	Error bounded by distortion, Outlier detector, Estimation Error, Best-fit piecewise partition
Li et al. [35]	Time series	Reduced time point	Error threshold	
Sharma [46]	Time series	Representative time point	Error threshold	
Chen et al. [40]	Time series	Representative time series	Window length, Compression rate	Temporal Edge-operator
Zhang et al. [45]	2-D Array	Raw data, Regression coefficients	Regression error bound	Regression error bound Multivariate DC
Azar et al. [57]	Float 2-Darray	Byte array	Error bound, Period	Compress M sensors Linear-Curve Fitting
Li et al. [35]	Time series	Curve coefficients: three (complete curve) four (incomplete curve)	Error threshold, Window size	
Kasirajan et al. [41]	Time series	Quantized value	Error bound	Reconstruction error bounded
Mendes et al. [59]	Time series	Linear, Order 2, Order 3, Exponential coefficients	Error threshold, Window size	
Klus et al. [60]	Time series	Representative time point	Error threshold	

5.2.2. Statistical inference

Statistical inference methods aim to infer the behavior of a given population through information provided by samples. It is the process of deducing properties using data analysis.

Li and Li [34] propose an approach that combines a sampling frequency control algorithm and a data compression algorithm by focusing on energy saving. The Precision Based Sampling and Transmit Algorithm (PBSA) dynamically adjusts the sampling frequency on sensing data. PBSA uses a linear regression model to predict the time interval used. The Data Compression Algorithm (DCA) is used when sampling frequency cannot be controlled. The CR obtained by DCA is low when a small error criterion is chosen.

A two-level data reduction approach is presented by Harb et al. [56]. The first level of compression occurs in the sensor node. The compression algorithm is based on the Pearson coefficient that represents the degree of correlation between two data sets. A Pearson threshold is employed as a compression criterion. A sample array is separated into subsets that are analyzed by the compression algorithm. A representative array is created to represent the original data. The second level eliminates redundant data generated by neighboring's nodes using cluster algorithms (EKmeans or TopK) in an intermediate node. A good selection of the set size and the Pearson criterion are indispensables because the execution time and the data quality depend on them.

Liu and Yu [39] propose the monotone increase compression method to calculate the time difference of arrival (TDOA) in Intelligent Transportation Systems (ITS). The proposal focuses only on this specific application. Data compressed by this method may be very lossy but give satisfactory results for calculating the TDOA.

5.3. Interpolation

The creation of a new data point using a range of a discrete set of known data points is defined as interpolation. The sensor signals could be seen as the values of a function. The values of a timeline are used as an independent variable in time-domain signals. When a signal is interpolated, the value of that function is estimated for an intermediate value. Interpolation is a good tool for DC. Sensor signals can be represented with approximate or representative values that allow having an estimate of the original signal with fewer samples. Table 8 presents the basic information of the interpolation approaches.

5.4. Linear interpolation

The simplest method of interpolation is by representing signals using a linear function. This method is not very accurate, but some approaches manage to control the resulting error.

Kui Zhao et al. [43] proposes the Distributed Swinging Door Trending (DSDT). DSDT is a modification of SDT [63], which uses a linear trend to represent some samples using error threshold criteria. The proposal allows compressing data on the sensor node or externally in a place named compressing center. The error threshold must be properly selected, which implies a previous analysis of the signal behavior by the system designer. The computational complexity of this mechanism is $O(1)$.

Pham et al. [37] propose a Top-Down Piecewise Linear Approximation with Minimum number of Line Segments (PLAMLiS). This approach uses an error bound to control the compression error. PLAMLiS analyzes a set of samples that are separated into subsets. Each subset has a linear representation that meets the error criteria selected. The computational complexity of this approach is $O(nlogn)$.

A spatiotemporal tree-structure linear approximation scheme is proposed by Wang et al. [52]. The proposal has two contributions: a bottom-up procedure to explore the best-fit piecewise partition for the sample set; and a scheme adaptable to heterogeneous sensors, various sampling rates, and outlier data. The proposal uses Tree-structured Optimal Pruning and linear regression to extract the slope and the intercept of a possible line that can be used to approximate the data set. The algorithm worst-case has a complexity of $O(Wlog2W)$.

Li et al. [35] propose to use the linear approximation to represent sensor data. Linear approximation represents the data by a line segment needing only two points to represent a set of samples. The approach uses a simple error bound: samples are released when the current sample has a greater error than the last sample. The mechanism has a complexity of O(1).

Lightweight temporal compression (LTC) is proposed by Sharma [46]. LTC is based on a piecewise linear approximation. This technique uses lower and upper limits (size of E) to evaluate when it is necessary to send the last point of a trend line. The proposal has an error control system more adequate than the one proposed by Li et al. [35]. LTC has a computational complexity of $O(1)$.

A non-threshold-based node level algorithm is proposed by Chen et al. [40]. The approach does not require any prior knowledge and predefined threshold. The basic idea of edge operator is used to design the Extension Temporal Edge-operator (ETEO). A convolution is done between ETEO and time series for getting a new sequence. The algorithm focus on reaching an objective CR that causes an uncontrolled error.

A self-Adaptive Regression-based Multivariate Data Wavelet Compression Scheme with Error Bound (AR-MWCEB) is proposed by Zhang et al. [45]. The proposal is designed to be used in sensor nodes that simultaneously collect data from different sensors. The proposal is based on the following three facts: a multivariate correlation exists, a spatial correlation does not exist, and the error is bounded. AR determine the number of data involved in the regression calculation. The algorithm decides between sending regression coefficients or raw data, considering the error and the application needs.

Klus et al. [60] propose the Direct Lightweight Temporal Compression (DLTC) method that is an improvement of the LTC presented in [52]. The advantages of the DLTC over LTC are latency reduction and a lower CE. It also reduces the number of algebraic operations at each algorithm integration.

The proposals presented in Kui Zhao et al. [43] and Wang et al. [52] have higher computational complexity than the others approaches of linear interpolation. Nevertheless, they get better results of CE and CR.

5.5. Nonlinear interpolation

Azar et al. [57] propose a fast error-bounded lossy compressor. It uses a Lightweight modification of SZ as a DC algorithm. The proposal focuses on devices that collect M data from N sources (e.g., a sensor, gyroscope coordinates). The collected 2-D array is converted to a 1-D array. SZ uses a best-fit curve-fitting model [64]. This model checks each data point in the 1-D array for analyzing whether it can be predicted using user-required error-bound criteria. This best-fit step employs three prediction models: prediction Preceding Neighbor Fitting (PNF), Linear-Curve Fitting (LCF), and Quadratic-Curve Fitting (QCF). The compressed data are rebuilt in an edge node. This node process the decompressed data using Feed-Forward Neural Network (FFNN). The goal is to analyze the effect of lossy compression on the performance of the machine and the deep learning models. The case study was the stress level of drivers. The approaches present good results in the case study. Two of its steps have an $O(N)$ of time complexity.

An Adaptive Data Compression Mechanism is proposed by Mendes et al. [59]. The mechanism represents a set of measurements by curve fitting. The data compressed are represented by functional model coefficients. The collected data can be represented by a linear, polynomial (degree 2 or 3), or exponential function. The resulting coefficients are calculated using the least-squares method. The approach requires two parameters, error threshold, and windows size. The system requirements are used to configure the windows size, limiting the achieved CR.

Li et al. [35] propose to use the cubic Bézier curve approximation to compress sensor data. Bézier uses three coefficients for describing a signal. The algorithm considers an error bound. When the error limit is not fulfilled, the curve is considered incomplete and needs four coefficients for being represented (three coefficients and the last point). This approach has a higher data reduction ratio and complexity of $O(N)$, being N the number of data points involved in the curve representation.

Kasirajan et al. [41] propose the Non-linear Adaptive Pulse Coded Modulation-Based Compression (NADPCMC). The idea is to represent the data as a non-linear relationship and use techniques from adaptive estimation theory to obtain an accurate estimate. The proposal has two stages: the estimation stage, which performs a non-linear adaptive estimation, and the quantization stage, which quantizes the difference between the actual value and the estimated value.

Table 9
Transform information.

Document	Input	Output	Parameters	Features
Bashlovkina et al. [48]	Array	Fuzzy vector	Number of samples, Number of membership functions	Inter-sensor prediction scheme
Chen et al. [58]	Vector	Compressed vector	Length, Threshold	Additional compression step
Liu and Yu [39]	Vector	Coefficients	Decomposition level, Number of samples	Increased flexibility

5.6. Transform

Transform procedure converts data from one domain (e.g. time) to another (e.g., frequency). This new domain can represent the signal with less information. The transformation domain is an important tool in DC. Data can be transformed to other domain that represents them into a small number of coefficients.

Table 9 illustrates basic information of the transform approaches.

Bashlovkina et al. [48] propose the Fuzzy Compression Adaptative Transform (FuzzyCat). The proposal is based on the Fuzzy Transform Compression (FTC). FuzzyCat adapts the transform parameters to the signal's curvature inferred from the time derivatives and increases the resolution of the F-transform whenever the signal exhibits high curvature. The curvature detection is done by monitoring the second derivative of the signal.

A Hierarchical Adaptive Spatio-Temporal Data compression (HASDC) is proposed by Chen et al. [58]. HASDC uses the Adaptive Threshold compression algorithm (ATCA). Discrete Cousin Transform (DCT) is used to explore the temporal compression by devices and Discrete Wavelet transform (DWT) for spatial compression in cluster heads.

Liu and Yu [39] propose the use of the wavelet packet analysis with the Fisher-information technique. This approach provides great flexibility.

6. Methods comparison

This section presents a brief qualitative comparison between the DC methods surveyed. According to the proposed taxonomy, the analyzed methods are initially discussed among those of the same category. Finally, the categories are compared with each other.

Hybrid approaches can achieve good results in terms of CR without increasing the CE. These approaches can switch between a lossless and a lossy mechanism depending on the application necessities and requirements.

Table 10 briefly reviews the advantages and limitations of hybrid methods.

AI methods entail greater computational complexity but are suitable for use in complex applications, in which other types of solutions with conventional methods cannot solve. However, in applications with low complexity, it is not recommended. Table 11 presents a review of the advantages and limitations of the AI approaches surveyed.

Interpolation approaches are generally less complex than other methods. They have a simple configuration process and present good results in low-complexity applications, such as in some existing environmental monitoring systems. Table 12 presents characteristics of interpolation approaches.

Table 13 illustrates important characteristics of transform methods. These methods are recommended for signals that can be compressed using a data domain other than the time domain, such as frequency.

Finally, Table 14 summarizes the advantages, limitations, and applications for each category of methods in the proposed taxonomy. The task of comparing techniques from different categories is challenging. Moreover, some techniques in the same category may have their specificities. Despite this, it is possible to point out some characteristics common to most techniques in each category.

With the constant increase in the complexity of IoT applications and the simultaneous increase in the computational capacity of IoT devices, hybrid and AI methods are likely to have the greatest potential for future research. For now, implementing these techniques in simple IoT devices is still challenging.

7. Conclusions

Since their definition in 1999, IoT systems have grown substantially without interruption. The number of interconnected objects exceeded the human population in 2011, and tens of billions are expected to be deployed by 2025. The many connected sensor devices associated with these IoT objects generate a large amount of data from different sources, which requires to be transmitted, stored and processed.

This massive amount of data requires a huge increase in infrastructure resources (e.g., networks, storage, computing capacity). However, today, the growth rate of data generated by sensors is much higher than the growth rate of the capacity to increase infrastructure resources. Lossy data compression techniques have proven to be an adequate solution to optimize the use of resources

Table 10
Characteristics of hybrid methods.

Document	Advantages	Limitations	Application
Ramijak et al. [55]	Correlate and uncorrelated versions	Dictionary storage, Parameter configuration	Multi-attribute IoT data streams
Chen et al. [42]	Simple configuration, Dynamic Error	Simple error bound mechanism	WSN
Kolo et al. [51]	Adaptive compression, SME as parameter	Two Huffman tables for each sensor	Sensor data monitoring
Mohamed et al. [44]	Easy configuration, Event detection, Lossless for important events	Requires samples with strong time correlation	Sensor data monitoring
Giorgi [53]	Zero-latency approach, Simple configuration	Low CR	Biomedical signals, WBSN
Alsalaet and Ali [47]	Simple configuration	Develop to a specific application, Low CR	Vibration measurements
Capo-Chichi et al. [38]	Easy implementation	Limited error bound mechanism	Weather sensors, Tiny devices
Li and Liang [49]	Unified lossless and lossy compression	Low compression performance	WSN, Environmental monitoring
Zhang and Li [36]	Good performance in tiny devices	Focus exclusively on vibration sensors	Vibration sensors
Liu and Yu [39]		Focus exclusively on ITS, High CE	ITS

Table 11
Characteristics of AI methods.

Document	Advantages	Limitations	Application
Liu et al. [54]	Energy optimization, Good CE control	Requires high computational resources, Complex implementation	Low error tolerance applications, Biomedical signals
Abu Alsheikh et al. [50]	Outlier detector, Temporal and spacial DC	Requires data with spatio-temporal correlation	WSN
Li and Li [34]	Sampling frequency control, Low computational complexity	Very sensitive to signal changes	Environmental monitoring, Telemetry
Harb et al. [56]	Elimination of redundant data	CR depending on array size	WSN
Liu and Yu [39]		Compressed signal may be unrecognizable, Focuses on a specific application	ITS

in industrial IoT systems, allowing to reduce the amount of infrastructure needed for sensor data streams, increasing these systems' economic and energy viability.

The research question for this Systematic Literature Review resulted in 30 approaches, which were assessed and discussed. A taxonomy was proposed in which the proposed techniques were classified into hybrid, artificial intelligence, interpolation, and transform. Some conclusions could be drawn from the analysis of these proposals.

Interpolation approaches have less computational complexity than other methods analyzed in the proposed taxonomy. These methods usually use threshold tolerance parameters that control the compression error. They are suitable for implementations that need real-time data compression results and for time-domain sensor measurements.

Hybrid methods use lossy and lossless data compression techniques. The lossy technique compresses the initial data with a residual error, and then a lossless technique improves the compression ratio without increasing the compression error. Depending on the techniques used, a problem with hybrid approaches is that they can have greater computational complexity.

Table 12
Characteristics of interpolation methods.

Document	Advantages	Limitations	Application
Kui Zhao et al. [43]	CE and CR performances, Error threshold (signal trend)	High computational complexity to LI	IoT
Pham et al. [37]	Simple configuration, CR performance	Primitive error threshold mechanism	IoT, Environmental monitoring
Wang et al. [52]	Outlier detector, Scheme adaptable to heterogeneous sensors	High computational complexity to LI	WSN
Li et al. [35]	Constant computational complexity	Primitive error threshold, Poor CR performance	Data monitoring
Sharma [46]	Constant computa–tional complexity, Error control follows signal trend	Integer resolution	Environmental monitoring
Chen et al. [40]	CR performance	CE is not controlled	Data monitoring
Zhang et al. [45]	Multi-variable DC (multiple sensors), Deliver raw or compressed data depending on the application requirements	Complex algorithm, require to meet three facts	Multi-sensor monitoring, WSN
Azar et al. [57]	Multi-variable DC (multiple sensors), Decompression by ML	Complex implementation	Multi-sensor monitoring, WSN, WBSN
Li et al. [35]	CR and CE performances	High computational complexity, Windows size can limit CR	Data monitoring, WSN
Kasirajan et al. [41]	Easy configuration, CE performance	Limited CR results	WSN
Mendes et al. [59]	Easy configuration, CE performance	Limited CR results	Smart Grids, WSN Advanced metering infrastructure
Klus et al. [60]	Constant computational complexity, Error control follows signal trend, Reduce algebraic operations over LTC	Integer resolution	Low-latency applications, Sensor-based wearables

Table 13
Characteristics of transform methods.

Document	Advantages	Limitations	Application
Bashlovkina et al. [48]	Adaptation of transform parameter by signal behavior, Inter-sensor predictor scheme	Overhead due to additional processing time	WSN, Biomedical signals
Chen et al. [58]		Slow response	WSN
Liu and Yu [39]	More detailed data analysis	Focus on a specific application	ITS

Transform methods convert data from one domain to another. This transformation can represent the signal with coefficients or fewer samples, showing good compression results. These methods need data vectors as inputs, requiring a reasonable number of measurements to give satisfactory compression results.

Finally, artificial intelligence methods generally have more implementation complexity and require a pre-training process in many cases. These implementations are well suited to approaches with many inputs and more complexity in the system data.

Table 14
Characteristics of the taxonomy methods.

Classification	Advantages	Limitations	Application
Hybrid	Good CR results, Potential for future research	High computational complexity	WSN, Biomedical signals
AI	Useful in problems with high complexity and low error tolerance, Potential for future research	Highest computational complexity, More complex implementation	Low-error tolerance applications, Environmental monitoring, WSN, Biomedical signals
Interpolation	Low computational complexity, Easy implementation, Fast deployment	Highly variable output signal, Not recommended for complex problems	Telemetry, WSN, Multi-sensor environment, Environmental monitoring
Transform	Good CR results	Slow response time, Requires prior knowledge of signal behavior	WSN, Biomedical signals

When analyzing lossy DC approaches, we observe that the proposed approaches generally do not take into account the effect that outliers can have on the result of compression. Generally, works on outlier treatment and data compression are proposed separately. An interesting future work would be to carry out a cross-analysis between scientific works on detection/treatment of outliers and data compression techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This study was supported by CAPES/Brazil (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior): PrInt CAPES-UFSC “Automação 4.0”.

References

[1] L. Coetzee, J. Eksteen, The internet of things - promise for the future? An introduction, in: 2011 IST-Africa Conference Proceedings, 2011, pp. 1–9.
[2] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: A survey, IEEE Commun. Surv. Tutor. (ISSN: 1553-877X) 16 (1) (2014) 414–454, http://dx.doi.org/10.1109/SURV.2013.042313.00197.
[3] B. Dorsemaine, J. Gaulier, J. Wary, N. Kheir, P. Urien, Internet of things: A definition taxonomy, in: 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, 2015, pp. 72–77.
[4] P. Suresh, J.V. Daniel, V. Parthasarathy, R.H. Aswathy, A state of the art review on the internet of things (IoT) history, technology and fields of deployment, in: 2014 International Conference on Science Engineering and Management Research, ICSEMR, 2014, pp. 1–8, http://dx.doi.org/10.1109/ICSEMR.2014.7043637.
[5] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. (ISSN: 0167-739X) 29 (7) (2013) 1645–1660, http://dx.doi.org/10.1016/j.future.2013.01.010.
[6] S. Pattar, R. Buyya, K.R. Venugopal, S.S. Iyengar, L.M. Patnaik, Searching for the IoT resources: Fundamentals, requirements, comprehensive review, and future directions, IEEE Commun. Surv. Tutor. (ISSN: 1553-877X) 20 (3) (2018) 2101–2132, http://dx.doi.org/10.1109/COMST.2018.2825231.
[7] I. Ud Din, M. Guizani, S. Hassan, B. Kim, M. Khurram Khan, M. Atiquzzaman, S.H. Ahmed, The internet of things: A review of enabled technologies and future challenges, IEEE Access (ISSN: 2169-3536) 7 (2019) 7606–7640.
[8] T. Alam, A reliable communication framework and its use in internet of things (IoT), 2018, pp. 450–456, CSEIT1835111| Received, 10.
[9] M. Wollschlaeger, T. Sauter, J. Jasperneite, The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0, IEEE Ind. Electron. Mag. 11 (1) (2017) 17–27, http://dx.doi.org/10.1109/MIE.2017.2649104.
[10] A.R. Biswas, R. Giaffreda, IoT and cloud convergence: Opportunities and challenges, in: 2014 IEEE World Forum on Internet of Things, WF-IoT, 2014, pp. 375–376, http://dx.doi.org/10.1109/WF-IoT.2014.6803194.
[11] P.V. Paul, R. Saraswathi, The internet of things — A comprehensive survey, in: 2017 International Conference on Computation of Power, Energy Information and Commuincation, ICCPEIC, 2017, pp. 421–426, http://dx.doi.org/10.1109/ICCPEIC.2017.8290405.
[12] G.G.M. de Jesus, R.D. Souza, C. Montez, A. Hoeller, LoRaWAN adaptive data rate with flexible link margin, IEEE Internet Things J. 8 (7) (2021) 6053–6061, http://dx.doi.org/10.1109/JIOT.2020.3033797.
[13] J. Azar, A. Makhoul, M. Barhamgi, R. Couturier, An energy efficient IoT data compression approach for edge machine learning, Future Gener. Comput. Syst. (ISSN: 0167-739X) 96 (2019) 168–175, http://dx.doi.org/10.1016/j.future.2019.02.005.
[14] B.R. Stojkoska, Z. Nikolovski, Data compression for energy efficient IoT solutions, in: 2017 25th Telecommunication Forum (TELFOR), 2017, pp. 1–4, http://dx.doi.org/10.1109/TELFOR.2017.8249368.
[15] H. Harb, A. Makhoul, C. Abou Jaoude, A real-time massive data processing technique for densely distributed sensor networks, IEEE Access 6 (2018) 56551–56561, http://dx.doi.org/10.1109/ACCESS.2018.2872687.
[16] K.P. Shravana, D.S.V. Veena, Review on lossless data compression using X-Matchpro algorithm, in: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology, RTEICT, 2017, pp. 1095–1100, http://dx.doi.org/10.1109/RTEICT.2017.8256767.

[17] M. Hosseini, A survey of data compression algorithms and their applications, 2012, http://dx.doi.org/10.13140/2.1.4360.9924.
[18] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. (ISSN: 1389-1286) 54 (15) (2010) 2787–2805, http://dx.doi.org/10.1016/j.comnet.2010.05.010.
[19] O. Mahdi, M. Mohammed, A. Jasim Mohamed, Implementing a novel approach an convert audio compression to text coding via hybrid technique, IJCSI (2013).
[20] C. Andrews, J. Davies, G. Schwarz, Adaptive data compression, Proc. IEEE 55 (3) (1967) 267–277.
[21] D.A. Lelewer, D.S. Hirschberg, Data compression, ACM Comput. Surv. 19 (3) (1987) 261–296, http://dx.doi.org/10.1145/45072.45074.
[22] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, K. Araki, Practical data compression in wireless sensor networks: A survey, J. Netw. Comput. Appl. (ISSN: 1084-8045) 35 (1) (2012) 37–59, http://dx.doi.org/10.1016/j.jnca.2011.03.001, Collaborative Computing and Applications.
[23] H. Kotha, M. Tummanapally, V. Upadhyay, Review on lossless compression techniques, in: J. Phys.: Conf. Ser., 1228, 2019, http://dx.doi.org/10.1088/1742-6596/1228/1/012007,
[24] D.G. Costa, L.A. Guedes, A survey on multimedia-based cross-layer optimization in visual sensor networks, Sensors (ISSN: 1424-8220) 11 (5) (2011) 5439–5468, http://dx.doi.org/10.3390/s110505439.
[25] E. Moghadas, J. Rezazadeh, R. Farahbakhsh, An IoT patient monitoring based on fog computing and data mining: Cardiac arrhythmia usecase, Internet of Things (ISSN: 2542-6605) 11 (2020) 100251, http://dx.doi.org/10.1016/j.iot.2020.100251.
[26] K. Dolfus, T. Braun, An evaluation of compression schemes for wireless networks, in: International Congress on Ultra Modern Telecommunications and Control Systems, 2010, pp. 1183–1188, http://dx.doi.org/10.1109/ICUMT.2010.5676532.
[27] J. Uthayakumar, T. Vengattaraman, P. Dhavachelvan, A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications, J. King Saud Univ. - Comput. Inf. Sci. (ISSN: 1319-1578) (2018) http://dx.doi.org/10.1016/j.jksuci.2018.05.006.
[28] J.D.A. Correa, A.S.R. Pinto, C. Montez, E. Leão, Swinging door trending compression algorithm for IoT environments, in: Anais Estendidos Do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais, SBC, Porto Alegre, RS, Brasil, 2019, pp. 143–148, http://dx.doi.org/10.5753/sbesc_estendido.2019.8650.
[29] T. Bose, S. Bandyopadhyay, S. Kumar, A. Bhattacharyya, A. Pal, Signal characteristics on sensor data compression in IoT - An investigation, in: 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON, 2016, pp. 1–6, http://dx.doi.org/10.1109/SAHCN.2016.7733016.
[30] A.Y. Tuama, M.A. Mohamed, A. Muhammed, Z.M. Hanapi, R.R. Mohamed, K.A.A. Bakar, Recent advances of data compression in wireless sensor network, J. Eng. Appl. Sci. 13 (21) (2018) 9002–9015.
[31] J. Uthayakumar, T. Vengattaraman, P. Dhavachelvan, A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications, J. King Saud Univ. - Comput. Inf. Sci. (ISSN: 1319-1578) (2018) http://dx.doi.org/10.1016/j.jksuci.2018.05.006.
[32] N. Kimura, S. Latifi, A survey on data compression in wireless sensor networks, in: International Conference on Information Technology: Coding and Computing, ITCC'05 - Vol. II, Vol. 2, 2005, pp. 8–13 Vol. 2, http://dx.doi.org/10.1109/ITCC.2005.43.
[33] B. Kitchenham, Procedures for Performing Systematic Reviews, Vol. 33, no. 2004, Keele University, Keele, UK, 2004, pp. 1–26.
[34] J. Li, J. Li, Data sampling control and compression in sensor networks, in: X. Jia, J. Wu, Y. He (Eds.), Mobile Ad-Hoc and Sensor Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-32276-4, 2005, pp. 42–51.
[35] Y. Li, S.W. Loke, M.V. Ramakrishna, Energy-saving data approximation for data and queries in sensor networks, in: 2006 6th International Conference on ITS Telecommunications, 2006, pp. 782–785, http://dx.doi.org/10.1109/ITST.2006.289016.
[36] Y. Zhang, J. Li, Wavelet-based vibration sensor data compression technique for civil infrastructure condition monitoring, J. Comput. Civ. Eng. 20 (6) (2006) 390–399.
[37] N.D. Pham, T.D. Le, H. Choo, Enhance exploring temporal correlation for data collection in WSNs, in: 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies, 2008, pp. 204–208, http://dx.doi.org/10.1109/RIVF.2008.4586356.
[38] E.P. Capo-Chichi, H. Guyennet, J. Friedt, K-RLE: A new data compression algorithm for wireless sensor network, in: 2009 Third International Conference on Sensor Technologies and Applications, 2009, pp. 502–507, http://dx.doi.org/10.1109/SENSORCOMM.2009.84.
[39] X.C. Liu, L. Yu, Development of geolocation data compression for transportation target identification, Transp. Res. Rec. 2105 (2009) 71–82, http://dx.doi.org/10.3141/2105-10.
[40] F. Chen, Y. Shen, J. Liu, F. Wen, Nonthreshold-based node level algorithm of data compression over the wireless sensor networks, in: International Conference on Signal Processing Systems, Vol. 2, 2010, http://dx.doi.org/10.1109/ICSPS.2010.5555286.
[41] P. Kasirajan, C. Larsen, S. Jagannathan, A new adaptive compression scheme for data aggregation in wireless sensor networks, in: 2010 IEEE Wireless Communication and Networking Conference, (ISSN: 1558-2612) 2010, pp. 1–6, http://dx.doi.org/10.1109/WCNC.2010.5506541.
[42] Y. Chen, N. Huang, Y. Chu, M. Li, R. Chang, C. Wang, Dynamic bounded-error data compression and aggregation in wireless sensor network, in: SENSORS, 2012 IEEE, (ISSN: 1930-0395) 2012, pp. 1–4, http://dx.doi.org/10.1109/ICSENS.2012.6411224.
[43] J. Kui Zhao, L. Shun Mu, H. Ouyang, P. Fei Zhu, Y. Kai Li, L. Yang, L. Jun Chen, In-network time-series data compression for electric internet of things, Appl. Mech. Mater. 241–244 (2012) 3213–3223, http://dx.doi.org/10.4028/www.scientific.net/AMM.241-244.3213.
[44] M.I. Mohamed, W.Y. Wu, M. Moniri, Adaptive data compression for energy harvesting wireless sensor nodes, in: 2013 10th IEEE International Conference on Networking, Sensing and Control, ICNSC, 2013, pp. 633–638, http://dx.doi.org/10.1109/ICNSC.2013.6548812.
[45] J. Zhang, K. Yang, L. Xiang, Y. Luo, B. Xiong, Q. Tang, A self-adaptive regression-based multivariate data compression scheme with error bound in wireless sensor networks, Int. J. Distrib. Sens. Netw. 9 (3) (2013) 913497, http://dx.doi.org/10.1155/2013/913497.
[46] R. Sharma, A data compression application for wireless sensor networks using LTC algorithm, in: 2015 IEEE International Conference on Electro/Information Technology, EIT, (ISSN: 2154-0373) 2015, pp. 598–604, http://dx.doi.org/10.1109/EIT.2015.7293435.
[47] J.K. Alsalaet, A.A. Ali, Data compression in wireless sensors network using MDCT and embedded harmonic coding, ISA Trans. (ISSN: 0019-0578) 56 (2015) 261–267, http://dx.doi.org/10.1016/j.isatra.2014.11.023.
[48] V. Bashlovkina, M. Abdelaal, O. Theel, Fuzzycat: A lightweight adaptive transform for sensor data compression, in: 2015 IEEE International Conference on Communication Workshop, ICCW, 2015, pp. 2756–2762, http://dx.doi.org/10.1109/ICCW.2015.7247596.
[49] Y. Li, Y. Liang, Temporal lossless and lossy compression in wireless sensor networks, ACM Trans. Sensor Netw. 12 (2016) http://dx.doi.org/10.1145/2990196.
[50] M. Abu Alsheikh, S. Lin, D. Niyato, H. Tan, Rate-distortion balanced data compression for wireless sensor networks, IEEE Sens. J. (ISSN: 1530-437X) 16 (12) (2016) 5072–5083, http://dx.doi.org/10.1109/JSEN.2016.2550599.
[51] J. Kolo, L. Minn Ang, K. Phooi Seng, S. Shanmugam, W.G. Lim, Energy-efficient adaptive data compression in wireless sensor networks, Int. J. Sensor Netw. 22 (2016) 229, http://dx.doi.org/10.1504/IJSNET.2016.080371.
[52] C. Wang, C. Yen, W. Yang, J. Wang, Tree-structured linear approximation for data compression over WSNs, in: 2016 International Conference on Distributed Computing in Sensor Systems, DCOSS, 2016, pp. 43–51.
[53] G. Giorgi, A combined approach for real-time data compression in wireless body sensor networks, IEEE Sens. J. (ISSN: 1530-437X) 17 (18) (2017) 6129–6135, http://dx.doi.org/10.1109/JSEN.2017.2736249.
[54] J. Liu, F. Chen, D. Wang, Data compression based on stacked RBM-AE model for wireless sensor networks, Sensors 18 (2018) 4273, http://dx.doi.org/10.3390/s18124273.

[55] D. Ramijak, A. Pal, K. Kant, Pattern mining based compression of IoT data, in: ACM International Conference Proceeding Series, 2018, pp. 1–6, http://dx.doi.org/10.1145/3170521.3170533.
[56] H. Harb, A. Makhoul, C. Abou Jaoude, A real-time massive data processing technique for densely distributed sensor networks, IEEE Access (ISSN: 2169-3536) 6 (2018) 56551–56561, http://dx.doi.org/10.1109/ACCESS.2018.2872687.
[57] J. Azar, A. Makhoul, M. Barhamgi, R. Couturier, An energy efficient IoT data compression approach for edge machine learning, Future Gener. Comput. Syst. (ISSN: 0167-739X) 96 (2019) 168–175, http://dx.doi.org/10.1016/j.future.2019.02.005.
[58] S. Chen, J. Liu, K. Wang, M. Wu, A hierarchical adaptive spatio-temporal data compression scheme for wireless sensor networks, Wirel. Netw. (ISSN: 1572-8196) 25 (1) (2019) 429–438, http://dx.doi.org/10.1007/s11276-017-1570-6.
[59] D.L.S. Mendes, A.F. da S. Veloso, R.A.L. Rabelo, J.J.P.C. Rodrigues, J.V. dos Reis Junior, An adaptive data compression mechanism for wireless sensor networks in the smart grid scenarios, in: 2020 5th International Conference on Smart and Sustainable Technologies, SpliTech, 2020, pp. 1–6, http://dx.doi.org/10.23919/SpliTech49282.2020.9243795.
[60] L. Klus, R. Klus, E. Lohan, C. Granell, J. Talvitie, M. Valkama, J. Nurmi, Direct lightweight temporal compression for wearable sensor data, IEEE Sensors Letters 5 (2) (2021) http://dx.doi.org/10.1109/LSENS.2021.3051809.
[61] M. Li, C. Lin, C. Chuang, R. Chang, Error-bounded data compression using data, temporal and spatial correlations in wireless sensor networks, in: 2010 International Conference on Multimedia Information Networking and Security, 2010, pp. 111–115, http://dx.doi.org/10.1109/MINES.2010.31.
[62] F. Marcelloni, M. Vecchio, A simple algorithm for data compression in wireless sensor networks, IEEE Commun. Lett. 12 (6) (2008) 411–413, http://dx.doi.org/10.1109/LCOMM.2008.080300.
[63] E.H. Bristol, Swinging door trending: adaptive trend recording, in: Proc. of the ISA National Conf., 1990, pp. 749–753.
[64] S. Di, F. Cappello, Fast error-bounded lossy HPC data compression with SZ, in: 2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS, 2016, pp. 730–739, http://dx.doi.org/10.1109/IPDPS.2016.11.