

Review Article

Time Series Compression for IoT: A Systematic Literature Review

Marcos A. de Oliveira ^{1,2} Anderson M. da Rocha,^{1,2} Fernando E. Puntel,² and Gerson Geraldo H. Cavalheiro²

¹Federal Institute of Education, Science and Technology Farroupilha, Santa Maria, RS 97050-685, Brazil

²Technological Development Center, Federal University of Pelotas, Pelotas 96010-610, RS, Brazil

Correspondence should be addressed to Marcos A. de Oliveira; marcosjr.sm@gmail.com

Received 27 September 2022; Revised 8 May 2023; Accepted 29 June 2023; Published 1 August 2023

Academic Editor: Qin Hu

Copyright © 2023 Marcos A. de Oliveira et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Time series are widely used to record information in applications developed with Internet of Things (IoT) devices, where sensors are used to collect large amounts of data. These devices, even with the exponential technological evolution of the last years, present limitations for the handling of this data given the enormous volume of data collected. Low storage or processing capacity and environments with restricted connection resources make it difficult to manipulate and transmit information. In this context, compression of information collected in the format of time series can be an alternative to reduce the amount of data handled by devices. However, given the peculiarities of IoT devices and applications, the compression process cannot be more costly than handling raw data. To identify the existing solutions to compress time series for IoT applications. In this paper, we present a descriptive systematic literature review on this topic. Based on a well-defined protocol, 40 papers were selected for review to analyze the strategies used, as well as performance and limitations. In this review, we aim to identify different approaches to solving the problem of data compression in the context addressed, as well as identifying directions for future research. The use of performance metrics in the reviewed papers was also reported in detail, to better understand how the authors compare their solutions to others. Additionally, the relationship between time series compression and machine learning (ML) techniques was highlighted. Being aware of the state-of-the-art in time series compression solutions for IoT can help us project future trends and challenges regarding this process and also to identify which algorithms, methods, and techniques can best be used in combination with ML models, the purpose inherent to the IoT context.

1. Introduction

Nowadays, data compression is a major challenge for computer scientists. With big data, problems regarding storing, communicating, and processing large amounts of data have increased. Furthermore, the last decade has seen a great evolution of devices based on the Internet of Things (IoT) concept and the increasing use of these devices for data collection [1]. In this context, data are collected by different sensors, manipulated by a wide range of devices with different architectures, and sent through different communication networks. One of the most commonly used formats in IoT applications is time series due to its property of storing temporal records around measured variables.

Time series has been widely used for storing sensor data [2], especially after the recent technological evolution in mobile and embedded computing, incorporated into the concepts of edge and fog computing. However, although the recent technological progress of these devices is notable, the large volume of data collected still presents itself as a challenge to be overcome by the applications that manipulate this data.

In the vast majority of applications with IoT architecture, data are collected at the edge layer, transmitted through the fog layer, sent to the cloud layer for some heavier processing, and later returned to the lower layers. During this cycle, it is necessary to store and transmit a large volume of data on devices with limited resources. For this, it is important that there are alternative ways to represent the collected data to

minimize the demand for resources [3]. Thus, it is essential to investigate alternatives that enable an efficient manipulation and use of these data, aimed at IoT applications, given the particularities of this context.

Temporal data compression is an inherent theme of computing, regardless of the era [4]. New data formats emerge new alternatives for manipulation as well, and with the technological evolution of recent years, especially in IoT devices, the number of solutions to this issue has increased. Thus, this knowledge must be gathered, organized, and categorized to help researchers in the development of alternative approaches, as well as to identify the major research gaps that still exist. Following the evolution of IoT devices and applications, as well as market requirements, many studies have been developed in recent years to find alternatives to compress time series in the IoT context, providing better use of available computational resources. Thus, to better understand this topic and catalog existing knowledge, we have performed a systematic literature review (SLR).

1.1. Related Work. Some works related to the main topic of this SLR are presented below. There are many works, such as reviews and surveys, related to data compression, however, few focused on time series and the IoT context.

Jayasankar et al. [5], the authors present a survey on data compression, dividing and comparing the solutions found according to the type of data (text, audio, image, and video). As major contributions, the authors point out important research directions, highlighting the trend toward the exploration of artificial intelligence (AI). Both when using compressed data in machine learning (ML) tasks and using AI techniques to compress data, the authors assume an increase in using this technology to seek improvements in aspects such as efficient representation, reduction of computational complexity, and memory management.

Concerning time series compression, Chiarot and Silvestri [6] presents a survey focused on this subject, classifying the algorithms based on how they work and creating a taxonomy. They divided the 15 found algorithms into dictionary based, functional approximation, autoencoders, sequential algorithms, and others, evaluating and comparing them on aspects such as adaptability, symmetry, and loss. The authors performed an analysis, based on metrics such as compression ratio and speed, and highlighted the positive and negative points of the solutions, summarizing the best application for each technique.

About the data measured by sensors, Wen et al. [7] presents a study on the compression techniques for smart meter big data. The focus of the work is on electrical measurements in smart grids, using the time series format, and they address some compression methods for this type of data in this context. They detail the operation of some algorithms and introduce the advantages and disadvantages of each compression method, comparing the results obtained with the methods applied to data in the context of smart metering. As the main conclusions of the article, the authors highlight the importance found in the compression experiments to avoid the communication overhead and storage needs of data centers. They also highlight the need to monitor the evolution of

application requirements to balance the tradeoff between compression rate, information loss, and efficiency.

Given the evolution in the IoT area in recent years at an exponential speed, no systematic reviews of this nature were found on this topic, especially using well-defined protocols for conducting and reporting results. In the same sense, given the expansion of research possibilities, recent works, which deal with current technologies, also do not cover all relevant gaps. Mainly due to the relative novelty of the IoT and the constant technological revolution, it is necessary to follow the evolution of time series compression solutions, as well as their application particularities. Thus, to our best knowledge, it is presented here this SRL, concerning time series compression for IoT applications, to bring together many developments in the last 7 years on this topic, due to the concentration of publications in this period, observed by preliminary research.

1.2. Contributions and Research Structure. This work aims to propose an investigation of alternatives to represent time series in a compressed way, to identify the means of reducing the volume of this data. Allied to this, we seek to identify alternatives suitable for the IoT context and that do not compromise the quality of information, in the direction of enabling the use of compressed data in the applications for which they were collected, usually ML activities, in model training and data classification. The main contributions of this systematic review are the following:

- (i) Summarize the main research published in the last 7 years regarding time series compression for IoT applications;
- (ii) Discuss the main solutions (algorithms, techniques, and strategies) found for time series compression for IoT;
- (iii) Discuss the main metrics used to evaluate the solutions proposed by the reviewed papers;
- (iv) Identify the state-of-the-art on time series compression solutions for IoT applications;
- (v) Highlight open perspectives and future research directions about time series compression for IoT.

This article is structured in six sections, as follows; Section Background provides the necessary background for understanding the review through an overview of the concepts; Section Research Methodology presents the procedures and artifacts of the SLR, according to each stage; Section Results detail the studies selected for the review and discuss them; in Section Review findings and discussions, a discussion of future research directions is presented based on this review, finally, Section Conclusion presents the conclusion of the paper.

2. Background

This section presents an overview of the concepts covered in the paper, to contextualize the reader in the review's scope and to help to understand the research findings.

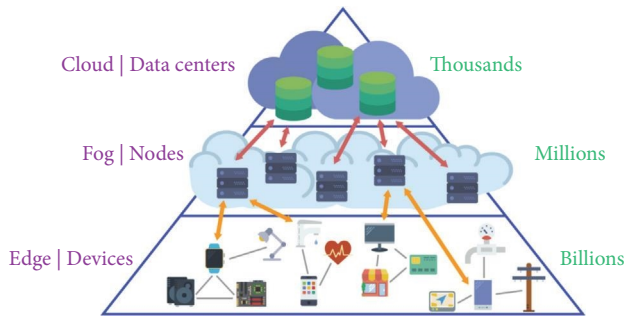


FIGURE 1: Distribution of IoT layers and devices (this image and other very similar representations are popular on the Internet. Al-Fuqaha [9] we found a description of the layers).

2.1. Internet of Things. The concept of the IoT can be defined as single identity objects operating in smart spaces capable of any time, anywhere, by anyone, and any service to communicate in different contexts [8]. Technological advances in recent years have expanded the connectivity and computational power of various devices, enabling them to integrate systems under the scope of IoT and increasing the number of IoT solutions in different contexts.

To better understand the operation of IoT applications, it is essential to know the application layers inherent to this context. Figure 1 presents a description of the layers, according to Al-Fuqaha [9]. IoT applications work in three layers: cloud, fog, and edge. The cloud layer, furthest from the end user, provided by a cloud server, is responsible for storing large volumes of data and heavy processing. The middle layer, called fog, is made up of devices with limited resources, which can perform a certain level of processing, being also responsible for communicating devices with the cloud server, as well as coordinating interactions between devices. Finally, the edge layer closest to the user usually encompasses data collection devices, such as sensors or mobile devices. The closer to the end user, the greater the number of applicable devices, represented in the figure in the order of thousands in the cloud layer, millions in the fog layer, and billions in the edge layer.

Connecting things with different resources to the internet means catalyzing the emergence of new applications and the use of devices such as sensors in contexts such as smart cities, healthcare monitoring, smart homes, and smart farms, among others. These applications are based on decision-making systems, supported by the main characteristic of IoT devices, which is to be an automated data collector. Data collection through sensors, combined with data processing techniques such as artificial intelligence, enables the automation of tasks such as monitoring objects and environments, detecting changes in sensed environments, acting in the environment through other devices, and other diverse possibilities of actions that can be triggered from sets of rules defined for a system.

As a result, challenges also emerge from this process: regulation, standardization, and security mechanisms are essential to guarantee the quality of the data collected [10]. The data provided by the objects can present imperfections

(sensor calibration), inconsistencies (out of order, outliers), and be of different types (generated by people, physical sensors, and data fusion). Furthermore, they can grow in volume in a short period, given their automated sensing nature. Thus, applications must be able to deal with these challenges in IoT environments [11].

In this scenario, the possibility of new applications is growing, as well as the need to monitor the challenges of connecting a wide range of things to the internet. Allied to this, it is important to always consider the restrictions of IoT devices, such as processing, autonomy, and limited memory, which highlight the importance of the continuous and updated study of these technologies.

2.1.1. Cloud. Cloud computing is an environment in which computing resources are used remotely over the internet, making it possible to use software or services that are not installed directly on the local computer. The technological evolution of recent years has made it possible to offer, through cloud services, infrastructure, and support for the development of various applications, such as IoT systems. Its widespread technologies make it possible for a customer to build their *private* cloud and exploit *public* ones. In this layer, there are cloud servers, where applications such as software, search tools, storage centers, and data processing are available.

Among several advantages, from a development perspective, cloud services allow developers not to spend efforts on issues such as network infrastructure [12]. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a set of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released [13]. Therefore, it is an environment with wide distribution and various forms of access, since it can be reached by other components of the application over the Internet.

2.1.2. Fog. Fog Computing is a concept that aims to distribute advanced storage, network, and management services closer to end users [14], sending information from IoT devices to Cloud Computing, forming a distributed and virtualized platform [15]. Fog Computing introduces a new architecture that brings processing to the data, while the cloud brings the data to the processing [16]. In this layer, devices such as small board computers are commonly used, due to their ability to perform a certain level of processing.

One of the main goals of fog computing is to reduce latency to enable ultra latency-sensitive applications. This type of application requires low end-to-end latency and, therefore, the components or processing units that make up such applications cannot be completely executed in the cloud. Besides communication latency, the problem of network congestion is also addressed here. This problem is quite common in cloud-centric solutions, as the cloud is several hops away from the data-producing devices and the data has to travel through different links that are shared by different users and applications [17].

2.1.3. Edge. Edge Computing is a computing model that allows you to store and process data closer to the end user.

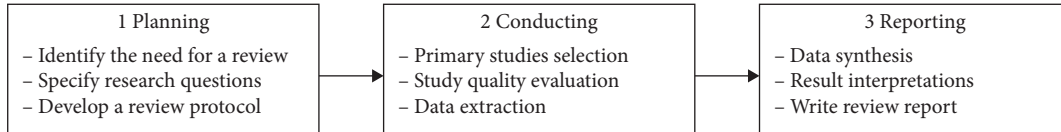


FIGURE 2: Systematic literature review process [26].

Edge Computing has emerged to deal with the demands for traffic and data processing, which have become increasingly voluminous and growing [18]. In general, Edge Computing emerged as a solution for cases where IoT devices are in situations of limited computing resources, such as poor connectivity. In this layer, the most common devices are sensors of different types, capable of collecting data from the environment where they are inserted.

In addition, Edge Computing makes it possible for part of the computation necessary for an application to be performed at the edge of the network, that is, on devices closer to the users' devices, such as routers, switches, and base stations. In this context, devices consume and produce data. They request resources and services from the cloud, but also perform computing on the cloud [19].

2.2. Time Series Compression for IoT Applications. Time series is applied in a wide variety of contexts, being, therefore, the subject of research for a long time in computer science. One of the main challenges in handling data in this format is often the large volume of data expressed by a time series. A large volume of data means a great need for computational resources, whether to store, manipulate, or transmit this data, which makes the investigation of time series compression relevant.

There are several strategies for data compression [20], many of them applied to time series. Methods to reduce the volume of data in a time series format can be used for general purposes or to preserve certain characteristics of the data sets, depending on the context in which these data are inserted. In some contexts, such as the IoT, it may be quite desirable to find a compression strategy that is capable of reducing the volume of data with guarantees that its quality is not compromised. The biggest challenge in this process is IoT devices, characterized mostly by the limitation of computational resources.

Driven by recent technological developments, applications that use IoT systems today are applied in several areas. One of the most prominent applications of IoT devices is in agriculture. The use of IoT sensors and systems in this area, for example, has enabled increases in productivity, cost reduction, and sustainable consumption of natural resources. In this environment, considerable equipment is used, causing a variety of data sources at high frequency and manipulated by very different devices. For this reason, many techniques have been developed over the years to balance the tradeoff between volume reduction, compression cost, and data quality. Balancing this equation is an important research focus for many researchers and this review meets this need, summarizing the state-of-the-art research on this topic.

3. Research Methodology

SLRs are an efficient way to synthesize prior knowledge on a topic of interest, addressing trends, challenges, and research gaps [21, 22]. Among the different types of SLRs, this paper presents a *Descriptive Review*, according to the typology presented in [23], to bring together the state-of-the-art solutions that deal with the challenge of time series compression in IoT applications. Therefore, for a descriptive review, the protocol chosen by the authors was defined by Kitchenham et al. [24], given its applicability to this type of review, its focus on software engineering, and its relevance in Computer Science. In a complementary way, to increase the reliability and robustness of the review presented in this paper, the RODA [25] protocol was also used. RODA is a computer science research guideline, which aims to help researchers from the conception and execution of the work to the presentation of results.

Thus, the SLR, according to Kitchenham et al. [24], has as major objectives: to synthesize existing evidence regarding a topic; identify gaps in current research to suggest future research; and provide a background to position recent research activities. The systematic review process followed comprises three stages [26]: *Planning*, *Conducting*, and *Reporting*, as shown in Figure 2. In the first stage, the planning, the review protocol is defined, which contains the motivation for the review, as well as the research questions, search string, research bases, and criteria for inclusion and exclusion of studies. In the second stage, the conducting, based on the previously defined protocol, the search and selection of the studies is performed, followed by the quality assessment and data extraction of the selected studies. In the third stage, reporting, the main report is constructed, a summary of the information extracted from the selected studies. Based on this report, research gaps and future research directions about the review can be observed. The details of the review steps are described in the next section of this paper. Each step of the SLR is described in a high level of detail, aiming to improve the reproducibility of this study.

Since it is a nontrivial process, which can be complex and lead researchers to analyze a large volume of data (according to the topic researched), several authors, such as [27], suggest the use of electronic tools, such as reference libraries, spreadsheets, and tools to automate the steps of the review, as a way to increase the reliability of data manipulation and, the credibility of the review. The main software used to perform this review was *Parsifal* (<https://parsif.al/>), a free online tool, designed to support researchers in conducting SLRs in the context of Software Engineering and chosen primarily for providing a shared online workspace, unlike other tools that work locally. This possibility was significant since this

review was carried out remotely by the authors, because of the health crisis caused by the coronavirus pandemic. *Parsifal* is a powerful tool, which helps in documenting the entire process, as well as allowing integration with reference management tools, such as *Mendeley* (<https://www.mendeley.com/>) and *Zotero* (<https://www.zotero.org/>), also used in this systematic review. We use over one reference manager due to different formatting in the references extracted from different research bases. In the process of exporting references from the research bases and importing them into *Parsifal*, the *BibTex* (<http://www.bibtex.org/>) format was used and, due to character encoding problems, we used *Mendeley* for some bases and *Zotero* for others, according to the output format of references in each case. Besides the aforementioned tools, *Google Sheets* (<https://www.google.com/sheets>) spreadsheets were also used, mainly for the extraction and recording of quantitative data, as well as the generation of graphics. Relevant details on the use of these tools at each stage will also be presented below.

3.1. Planning. The planning of a systematic review begins with the definition of an artifact that will serve as a guide for the procedure of the following steps, called a *Protocol*. The protocol contains fundamentally the definition of the objective, the PICOC form, the research questions, the keywords used in the search, the search string, the databases used for research, and the inclusion and exclusion criteria for studies. For the protocol construction, the concepts presented in [26] were followed, since they are the same definitions used for the construction of the *Parsifal* tool, for recording the information and procedures of the systematic review.

3.2. Objective and PICOC. Based on the motivations previously presented, the main objective of the systematic review was: *Identify the state-of-the-art in time series compression solutions for IoT applications.*

According to the literature, PICOC stands for *population, intervention, comparison, outcomes, and context*. For this work were defined: *population* as published studies about time series compression; *intervention* as solutions for time series compression; *outcome*, identification of time series compression solutions for IoT applications; and *context*, the primary studies about IoT applications using time series; *the comparison* was not applicable in this context.

3.3. Research Questions. Research questions help to specify the focus for the selection of primary studies, as well as data extraction and analysis. They were constructed to provide a better understanding of the selected papers, as well as to guide the authors in reporting an overview of the topic. Besides mapping existing solutions, and classifying and understanding them, a question related to agriculture was also included because it is a relevant work in front of the research group. Because the institutions of the authors involved in this review have many agricultural research groups, the results of this review can be used for the development of new case studies, such as the works reported in published papers [28, 29].

Hence, the following are the research questions described in this study:

RQ1: *What solutions are there for time series compression in IoT applications?*

RQ2: *How can these solutions for time series compression in IoT applications be classified?*

RQ3: *What metrics are used to evaluate the performance of the proposed solutions?*

RQ4: *How are the proposed solutions compared?*

RQ5: *Which of these solutions can be used in machine learning?*

RQ6: *What factors motivate the use of time series compression solutions in IoT applications?*

RQ7: *In what context were these solutions developed?*

RQ8: *Which of these solutions were developed in agriculture?*

3.4. Keywords, Search String, and Databases. The keywords were extracted from the previous definitions, such as research questions, population, intervention, and context. They were used to build the search string. These were the keywords (K) and synonyms (S) defined for this review:

K: compression—S: *compaction, reduction*

K: internet of things—S: *internet-of-things, IoT*

K: time series—S: *temporal series*

The search string was formed by the keywords and their synonyms and used as input for the search in the databases, with the view to return the studies related to the theme of the review. Good search results go through the definition of an appropriate search string. At this stage, tests were performed in a cyclical and revised way by all authors, for each combination of keywords that appeared mostly in the control articles. Control articles were the works shown by experienced researchers in the area and the most present in the related work. In general, terms were observed for the quality and quantity of the returned articles, as well as their relevance for the review, and then used the best combinations of words to define the final string. Excellent use of Boolean AND and OR operators was essential to separate keywords and synonyms to improve search research results. The search string defined for this review is shown below:

Search string: (“*time series*” OR “*temporal series*”) AND (“*compression*” OR “*compaction*” OR “*reduction*”) AND (“*internet of things*” OR “*internet-of-things*” OR “*IoT*”).

The defined search string was used in the following databases, chosen based on their relevance to the area of Computer Science:

ACM Digital Library: <http://portal.acm.org>

IEEE Digital Library: <http://ieeexplore.ieee.org>

Scopus: <http://www.scopus.com>

SpringerLink: <https://www.springer.com>

3.5. Inclusion and Exclusion Criteria. The study selection criteria aim to identify the primary studies that provide direct evidence on the research questions [26]. For reducing the likelihood of bias, the selection criteria decided upon during the protocol definition were also refined during the review process.

Regarding the period considered for this review (after 2016), during the calibration of the protocol, a low volume of publications on the topic outside this range was observed. This decision is justified because the dissemination of IoT devices has been concentrated in recent years, as a result of technological evolution and, consequently, a reduction in the cost of the devices.

In this way, the inclusion and exclusion criteria defined for this review are listed below.

3.5.1. Inclusion Criteria

- CI1: *articles returned from string search*
- CI2: *articles that present a time series compression solution*
- CI3: *articles using compressed data in machine learning applications*

3.5.2. Exclusion Criteria

- CE1: *articles published before 2016*
- CE2: *duplicate articles in search bases*
- CE3: *articles with secondary or tertiary studies*
- CE4: *articles with lecture notes, conference reviews (not research papers), and short papers (less than 4 p.)*
- CE5: *articles that do not use time series as data format*
- CE6: *articles with research focused on training and classifying machine learning models*
- CE7: *articles that do not present implementation or experiments of the solution proposed*
- CE8: *articles without access to the full text*

3.6. Quality Assessment Checklist. Besides defining the criteria for the selection of studies, the next step was to define some questions to assess the quality of the studies returned from the search. These questions help to give weight to the most relevant studies for review and filter those that are not. Questions QA1–QA9 aim to help researchers to observe aspects such as the focus and motivation of the work if experiments were carried out to validate the proposals, which metrics were used, and the context in which the solution was developed. A specific question (QA8) was included here regarding the agricultural context, as it is a topic of special attention and future interest of the research group. The evaluation questions used in this review are listed below.

- QA1: *Does the study present a solution for time series compression?*
- QA2: *Is the solution focused on IoT applications?*
- QA3: *Are the purpose and functioning of this solution presented (algorithm steps, source code)?*

QA4: *Do the authors present the metrics used to evaluate the proposed solution?*

QA5: *Do the authors discuss the limitations of the proposed solution?*

QA6: *Does the article present an experiment to show the application of the proposed solution?*

QA7: *Do the experiments presented include comparisons with other solutions?*

QA8: *Do the experiments include the application of the proposed solution in the agricultural context?*

QA9: *Did the authors explore the proposed solution in machine learning applications?*

A value was assigned to each question and a minimum grade was set, so a study must achieve that grade to be considered in the review. The possible answers to the questions were *yes*, *partially*, or *no*, with each answer being graded with a score of 1.0, 0.5, and 0.0, respectively. Therefore, the score for each study could range from 0.0 to 9.0, once nine questions were defined. A cutoff score of 5.5 was also defined, based on the best practice guidelines.

3.7. Data Extraction Form. For the data extraction step, a form was built with questions to record the information in a structured way. The Parsifal tool also allows the definition of a specific data type for the answers (string, boolean, and date) and the configuration of multiple choice questions. Here, some settings available in Parsifal were used, since extracting data in this way enables the generation of some reporting artifacts, such as graphics, in the tool. This is also a stage where rounds of tests were carried out, to identify the suitability of the form, according to the desired data to be extracted. The questions of the data extraction form used in this review are listed below.

- DE1: *Authors and Institution information*
- DE2: *Publication Date and Country*
- DE3: *Do the article authors propose a time series compression method?*
- DE4: *Name of the proposed solution*
- DE5: *Solution type*
- DE6: *Objective of the proposed solution*
- DE7: *Is the solution applied to the IoT context?*
- DE8: *Does it present experiments?*
- DE9: *The metrics used in the evaluation/analysis*
- DE10: *Does it show an improvement compared to other methods?*
- DE11: *Database privacy and URL (if available)*
- DE12: *In what context was the method developed?*
- DE13: *What is the objective of applying the method in the agricultural context? (if applied)*
- DE14: *Is Compression combined with machine learning? What algorithms are used? (if applicable)*

3.8. Conducting. The stage of conducting the SLR refers to performing the search and selection of articles related to the

TABLE 1: The number of excluded articles at each step and the selected ones for the review.

Database	Search	Excluded papers					Accepted	QA cut	Selected
		CE1	CE2	CE3	CE4	CE5, CE6, CE7, CE8			
ACM	904	60	18	147	158	489	32	19	13
IEEE	125	7	18	1	2	69	28	13	15
Scopus	180	7	68	6	23	60	16	11	5
Springer	603	47	1	125	5	413	12	5	7
Total	1,812	121	105	279	188	1,031	88	48	40

topic, based on the definitions presented in the previous section in the *Protocol*. The search for articles was performed using the search string and predefined databases. The articles returned by the search were then analyzed and labeled as excluded or accepted for review, according to the inclusion and exclusion criteria, also previously defined. Accepted articles were then evaluated according to the *Quality Assessment Checklist*, to assign a grade to each article. Those who had a score higher than the cutoff point went to the data extraction stage, where the *Data Extraction Form* was then used.

3.9. Search and Filtering Process. The search for studies was carried out through each database search engine, using the *CAFe* [30] access via *CAPES Journals* [31]. The final search was carried out in January 2023, therefore the review encompasses the period of full years from 2016 to 2022. After the search, the article data were exported in *BibTex* format, from the databases to reference management tools, such as *Mendeley* and *Zotero*, for character normalization, due to differences between the formats extracted from different databases. Once normalized, the data were imported into *Parsifal*, so that the review could then proceed with this tool. In *Parsifal*, after analyzing the data from the articles, they were labeled as accepted or rejected, as well as indicating the inclusion or exclusion criteria related to the label.

The search string is usually composed of traditional terms in the academic world to ensure that it returns the main works on the topic. The search for articles has, as an initial result, numerous returning articles. This also happens because of the high number of articles stored and available in the chosen databases. Hence, the objective of each phase of study selection was to restrict the search scope, identifying those studies that deal with the research problem of this review. Table 1 then presents the data for each step of this review, showing the number of articles returned at the beginning of the review, as well as the number of articles excluded after applying each exclusion criterion (CE columns) and the QA form, up to the number of selected papers. Also, Table 2 presents the title and an identification number of all selected papers, in alphabetic order.

First, *Inclusion and Exclusion Criteria* were used. The *Inclusion Criteria* were used to delimit the search scope. Then, the *Exclusion Criteria* were applied to specialize the review. The initial filtering consisted of checking the first four exclusion criteria (CE1, CE2, CE3, and CE4), individually and sequentially, referring to the date of publication, detection of duplicate returns, and article category, where

articles that did not meet these criteria were excluded. After this initial review, the filtering continues through the analysis of other information about the articles, such as the title and abstract, and, sometimes, when necessary, a dynamic reading was performed. Then, were used the other exclusion criteria (CE5, CE6, CE7, and CE8) to filter the articles. To reduce the possibility of misunderstandings about the purpose and contributions of the papers, the selection of studies was peer reviewed, randomly. Thus, two authors performed the review and classification of the paper and, in case of disagreement, the other authors also performed a review and the final opinion was discussed among all authors. After this step, the remaining articles (88) were accepted for review.

3.10. Quality Assessment and Data Extraction. Accepted articles (88) were then evaluated using the *Quality Assessment Checklist*, also in *Parsifal*. This form was defined in the protocol, as well as the cutoff point. The number of articles excluded at this stage for not reaching the cutoff point is also stated in Table 1 (column QA Cut). Those studies evaluated above the cutoff point (40) were then selected for the data extraction step, as well as for detailed review and full reading, as listed in Table 2.

The data extraction step was also performed via *Parsifal*. In this phase, the previously defined *Data Extraction Form* was used. All information was stored in the online tool and, after the process, it was extracted in an electronic spreadsheet format. Both the quality assessment and data extraction were peer reviewed in parallel rounds by the authors, and after that, the reviews were merged to guarantee the reliability of the step.

4. Results

In this section, we discuss the results of the performed SLR, presenting a summary of the articles selected for the review, their characteristics, and pointing out differences between the scope, context, and functioning of each solution. Furthermore, we discuss the research questions listed in the previous section are answered based on the findings of this process.

4.1. References Synthesis. This synthesis brings together the key ideas from each paper selected that were relevant to this study. Table 3 presents a summary of information of selected articles. In this table, each paper is referenced by an identifier code (column *Id/Ref*), the country of researchers and institutions (column *Ctry*), the solution type (column *Type*), as well as the name of the proposed solution (column *Name*).

TABLE 2: Selected articles for review after quality assessment and scores.

Id	Title	Sc.
A1	A multitier data reduction mechanism for IoT sensors [32]	6.0
A2	A multivariate-time-series-prediction-based adaptive data transmission period control algorithm for... [33]	7.5
A3	A nearest neighbors based data filter for fog computing in IoT smart agriculture [34]	7.0
A4	A novel multiresolution representation for time series sensor data analysis [35]	6.0
A5	A randomly accessible lossless compression scheme for time-series data [36]	6.0
A6	Accuracy-aware data reduction for Internet of Things [37]	6.0
A7	Accurate and fast time series classification based on compressed random Shapelet Forest [38]	7.5
A8	Adaptive multivariate data compression in smart metering Internet of Things [39]	6.0
A9	Adaptive segmentation of streaming sensor data on edge devices [40]	6.0
A10	Agile lossless compression algorithm for big data of solar energy harvesting wireless sensor network [41]	6.0
A11	CORAD: correlation-aware compression of massive time series using sparse dictionary coding [42]	6.0
A12	Data reduction based on machine learning algorithms for fog computing in IoT smart agriculture [43]	8.0
A13	Decomposition-based approximation of time series data with max-error guarantees [44]	6.0
A14	Direct lightweight temporal compression for wearable sensor data [45]	7.0
A15	Embedded machine learning-based data reduction in application-specific constrained IoT networks [46]	7.0
A16	Energy-efficient sensory data gathering based on compressed sensing in IoT networks [47]	6.0
A17	Energy-efficient two-layer data transmission reduction protocol in periodic sensor networks of IoTs [48]	6.0
A18	Extract, compress and encode: LitNet an efficient autoencoder for noisy Time-Series data [49]	6.5
A19	Fog computing application of cyber-physical models of IoT devices with symbolic approximation algorit... [50]	6.5
A20	GRAIL: efficient time-series representation learning [51]	7.5
A21	Integration of IoT streaming data with efficient indexing and storage optimization [52]	7.0
A22	LFZip: lossy compression of multivariate floating-point time series data via improved prediction [53]	7.5
A23	Living on the edge: data transmission, storage, and analytics in continuous sensing environments [54]	7.5
A24	Low-latency analytics on colossal data streams with summary store [55]	6.5
A25	Near lossless time series data compression methods using statistics and deviation [56]	7.0
A26	Plato: approximate analytics over compressed time series with tight deterministic error guarantees [57]	6.5
A27	Powerstrip: high-performance compression for energy data [58]	6.0
A28	Robust IoT time-series classification with data compression and deep learning [59]	7.0
A29	Sprintz: time-series compression for the internet of things [60]	7.0
A30	Streaming piecewise linear approximation for efficient data management in edge computing [61]	6.5
A31	System identification at the extreme edge for network load reduction in vibration-based monitoring [62]	6.5
A32	The embedded IoT time series database for hybrid solid-state storage system [63]	6.0
A33	The Internet of Things as a deep neural network [64]	8.0
A34	Titchy: online time-series compression with random access for the Internet of Things [65]	7.0
A35	Toward a standard time-series representation for IoT based on CBOR templates [66]	7.5
A36	Transfer learning autoencoder used for compressing multimodal biosignal [67]	7.0
A37	TSCache: an efficient flash-based caching scheme for time-series data workloads [68]	6.5
A38	Two-level energy-efficient data reduction strategies based on SAX-LZW and hierarchical clustering for ... [69]	6.5
A39	Use of dynamic perceptually important points for data reduction in IoT [70]	7.5
A40	Vibration edge computing in maritime IoT [71]	7.0

Moreover, a brief explanation about the solution is presented (column *Proposed Solution*), along with the main goal of the solution (column *Goal*), the specific context in which the solution was designed (column *Context*) and whether the solution was combined with some ML technique and, if not, which one (s) (column *ML Alg*).

Regarding the type of solution (column *Type*), the papers were categorized into *Algorithm (ALG)*, when a new algorithm for time series compression was proposed; *Technique (TEC)*, when a technique was proposed, the use of over one algorithm for compression; *Extension (EXT)*, when the paper presents an extension to an existing approach; and

Combination (CMB), when the paper presents the combination of traditional time series compression approaches. TEC and CMB, at first, can be interpreted in a similar way, however the specialization between the two categories was to identify original works that propose a new compression technique, separating them from those that only use existing techniques, combining them in different ways.

Basic information for the identification of roles and a summary of their objectives are presented, highlighting some operating characteristics. The geographic data about the publications were extracted to map the origin of the researchers and institutions around the world that are publishing works related to the

TABLE 3: Summary information of selected articles.

Id/Ref	Ctry	Type	Name	Proposed solution	Goal	Context	ML Alg.
A1 [32]	CHN	TEC	Multi-tier DR	Data reduction mechanism (IoT nodes)	Reduce energy and storage	Agriculture	-
A2 [33]	USA	ALG	PBA-TPC	Multivariate TS data encoding	Reduce energy and transmissions	Smart meter	CNN
A3 [34]	BRA	CMB	NN-based DF	Approach to collect and filter data (fog)	Reduce storage and transmission	Agriculture	k-NN
A4 [35]	CHN	EXT	MPLR-IDP	Dimensional reduction for sensor TS	Multi-resolution PLR from TS	General	-
A5 [36]	DNK	TEC	RALCS-TSD	Low-cost access compression scheme	Avoid data decompression	General	-
A6 [37]	IRL	ALG	AA-DPIP	Accuracy-aware TS data reduction	Use context ML/DL feedback	Industry	LTSM
A7 [38]	CHN	CMB	CRSF	Fast and accurate TSC algorithm	Use SAX to compress shapelets	General	Random Forest
A8 [39]	IND	ALG	AMDC	IoT compression scheme	Reduce data dimensionality	Smart meter	-
A9 [40]	POL	ALG	C ² -CCSB	Cubic splinelet-based compression	Streaming data segmentation	General	-
A10 [41]	SAU	ALG	ALC-SEH	Lossless compress. for solar energy data	Fast decompression and queries	Smart meter	-
A11 [42]	CHE	ALG	CORAD	Compression exploiting correlation	Eliminate redundancy	General	-
A12 [43]	BRA	CMB	DR-Fog-SA	Data reduction for fog computing	Data storage in smart agriculture	Agriculture	Several
A13 [44]	AUS	ALG	DBA/DBAIS	Decomposition-based compression	Guarantee maximum error	Industry	-
A14 [45]	FIN	TEC	DLTC	Lossy compression method for TS	Energy-efficient data transmission	Wearables	-
A15 [46]	IND	EXT	ML-DR	ML-based IoT data reduction scheme	Reduce data transmissions	General	Several
A16 [47]	CHN	CMB	CCS&ASTGCN	Data gathering mechanism (sensor)	Energy-efficient data transmission	General	-
A17 [48]	IRQ	TEC	ETDTR	Data reduction scheme (aggregator)	Energy-efficient data transmission	Smart meter	-
A18 [49]	DEU	TEC	LitNet	Efficient autoencoder for Noisy TS	Multi-level TS feature extraction	Industry	CNN
A19 [50]	PRK	EXT	BOSS VS	Symbol TS approximation	Efficiente TS Classification	Industry	Several
A20 [51]	USA	CMB	GRAIL	Framework to build compact TS	Preserve properties (function)	General	-
A21 [52]	AUS	TEC	RNBitBlocking	Lossless TS compression framework	Data search without full decompr	Industry	-
A22 [53]	USA	CMB	LFZip	Error-bounded lossy compressor	Guarantee a maximum error	General	ANN
A23 [54]	USA	ALG	Spinneret	Hypersketching TS stream algorithm	Reduce storage and transmissions	General	Random Forest
A24 [55]	KOR	CMB	SummaryStore	Approximate TS store for analytics	Query accuracy and cost saving	General	-
A25 [56]	IND	TEC	TSDC-SDE	TS compression method	Use statistical and deviation	General	-
A26 [57]	USA	TEC	Plato	Fast approximate analytics on TS	Process and store compressed TS	General	-
A27 [58]	USA	ALG	Powerstrip	Nearly lossless compression algorithm	Exploit energy-data properties	Industry	-
A28 [59]	FRA	CMB	Squeeze	Error-bounded compression approach	Optimize compr./quality tradeoff	Wearables	DNN
A29 [60]	USA	ALG	Sprintz	TS compression for low-power devices	Reduce data without lose quality	General	-
A30 [61]	SWE	EXT	SPLA	Streaming approximation technique	Reduce streamed data	General	-
A31 [62]	USA	EXT	eS-TSOR	Sysid models at the extreme edge	Reduce network congestion	Industry	-
A32 [63]	CHN	TEC	TS-NSM	Embedded IoT TS Database	Redundancy elimination	General	-
A33 [64]	SWE	TEC	CEML-IoT	Distributed ML compression framework	Reduce data transmission	General	Autoencoder
A34 [65]	DNK	TEC	Titchy	IoT TS compression method	Reduce latency and retrieval cost	General	-
A35 [66]	FRA	EXT	VTS/MTS	TS representation for IoT	Explore CBOR templates	Industry	Autoencoder
A36 [67]	IND	TEC	TIL M-CDAE	TL-based autoencoder compression	Reconstruction from latent data	Biosignal	-
A37 [68]	USA	CMB	TSCache	Flash-based cache system for TS data	Adaptive time-aware cache policy	General	-
A38 [69]	IRQ	TEC	CBDR	Two-level data reduction approach	Reduce data at node and gateway	Smart meter	-
A39 [70]	IRL	EXT	DPIP	Delay-tolerant data reduction algorithm	Use important points to reduce	General	LTSM
A40 [71]	GBR	TEC	VEIoT	Real-time dynamic intelligent reduction	Reduction without lose sensitivity	Maritime	-

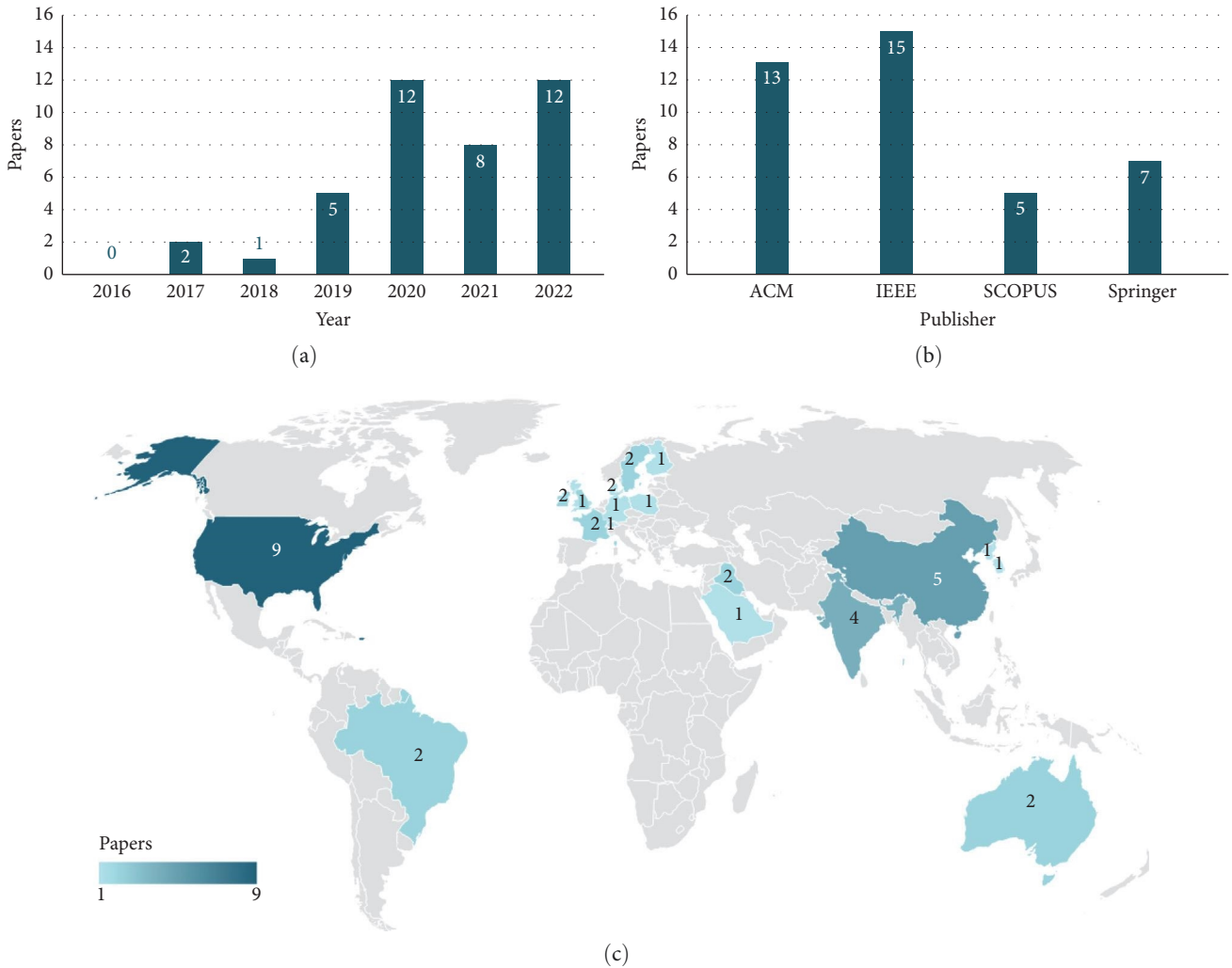


FIGURE 3: Distribution of selected papers by (a) year, (b) publisher, and (c) country.

topic. With this information, we seek to provide opportunities for researchers to identify groups for collaboration in future work. Figure 3 presents the distribution of publications of the papers selected for review by year, publisher, and country. In addition, the extraction of data related to the context of the proposed solution aims to verify if there is any predominant context regarding the development of solutions for time series compression, given the wide use of this data format.

4.2. Answers to the Research Questions. The SLR was conducted to identify papers presenting answers to the research questions proposed in this work. The answers found are presented below.

To answer the first question (RQ1: What solutions are there for time series compression in IoT applications?), 40 different solutions focused on time series compression with a focus on IoT applications. Although three solutions were not developed specifically for the IoT environment (A5, A26, and A37), they were also considered in this review as they apply to generic time series. These solutions are listed in Table 2, indexed in alphabetical order. Figure 3 complements the answer to this question through a temporal and geographical

distribution of the papers. From the figure, it is possible to observe that the number of published papers has been increasing, which shows the importance of the topic today. Also, the largest concentration of authors is in the United States and the publisher with the most papers published on this topic is the IEEE.

Additionally, to understand the relationships that exist or not between the works, Figure 4 shows the network of connections between authors of the 40 selected papers. To generate the graph, the VOSviewer (<https://www.vosviewer.com/>) was used. This software is a tool for constructing and visualizing bibliometric networks. Data from bibliographic references of the 40 selected papers were extracted in RIS (research information systems) format, and used as input for the tool. Based on the graph, it is noted that the relationships between authors are limited to their research groups and immediate research partners, with no connection between authors of two different papers.

Answering the second question (RQ2: How can these solutions for time series compression in IoT applications be classified?), the solutions were analyzed and grouped according to their nature. The 40 solutions were classified

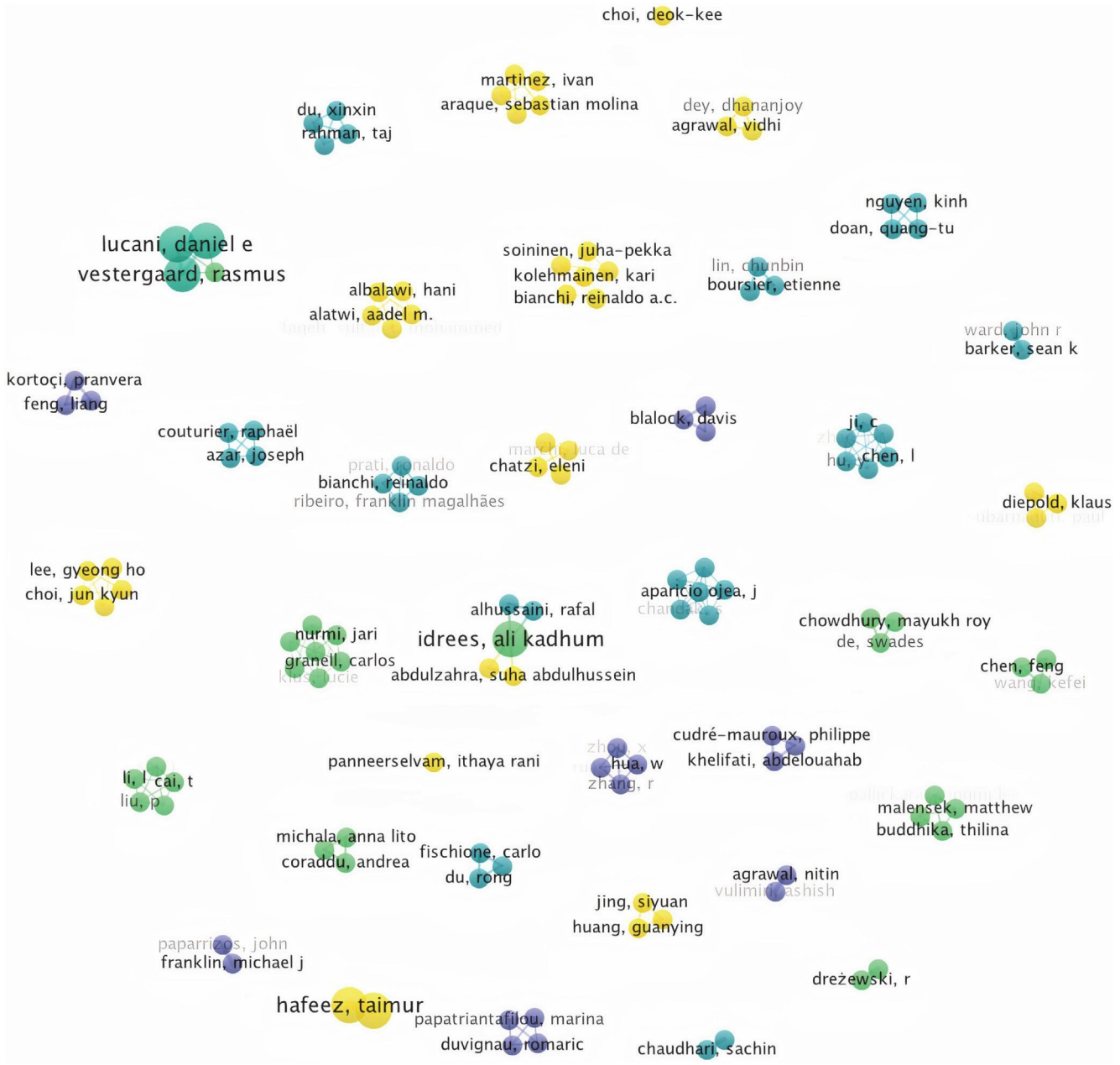


FIGURE 4: Connections between authors of the 40 selected papers.

into: *Algorithm* (10), *Technique* (14), *Extension* (7), or *Combination* (9). This classification is important for a correct performance evaluation of the proposed solution. Furthermore, this helps researchers to understand why and how a solution was developed, and how it can be compared with others of the same nature. Another important aspect is the application layer for which the solution is made. The identification of the application layer is an analysis that complements the classification of solutions. Figure 5 helps to understand the distribution of applications by type and by a layer of an application. It is possible to observe that, while there is a balanced distribution of solutions by type, in terms of the application layer, more than half of the solutions are

aimed at the cloud layer. This is justified by the difficulty of implementing edge solutions, which shows an interesting research gap.

To answer the third question (RQ3: What metrics are used to evaluate the performance of the proposed solutions?), all metrics used in all articles were tabulated. Figure 6 presents the number of uses of the 16 different metrics identified. From this, it is possible to highlight the three most used metrics, consistent with the review: *Compression Ratio*, *Accuracy*, and *Runtime*.

In addition, variations were observed regarding the use of the same metric or coefficient for different purposes. For example, the use of the same index as the accuracy rate in

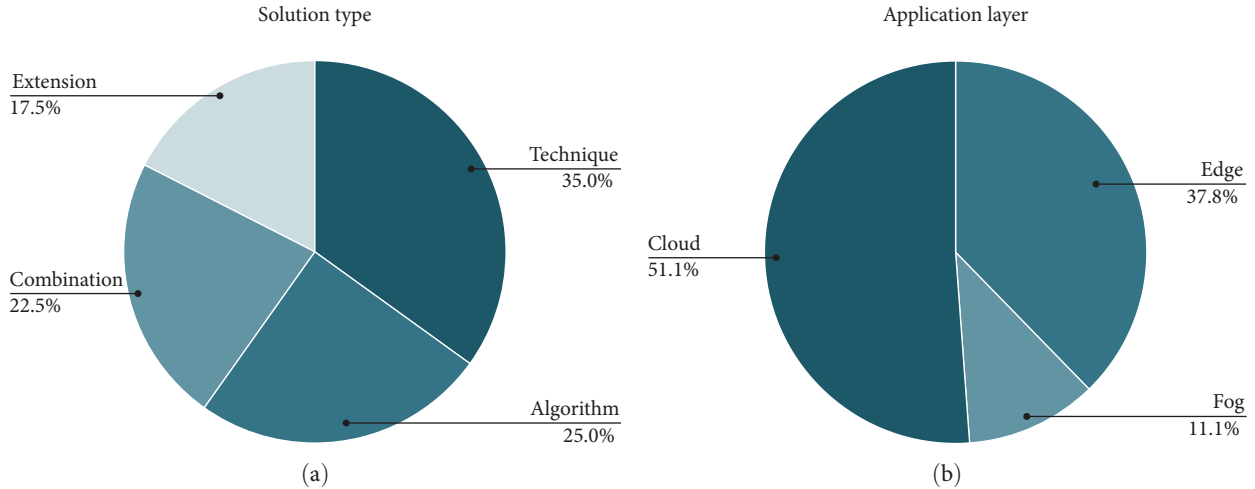


FIGURE 5: Distribution of papers by (a) solution type and (b) application layer (RQ2).

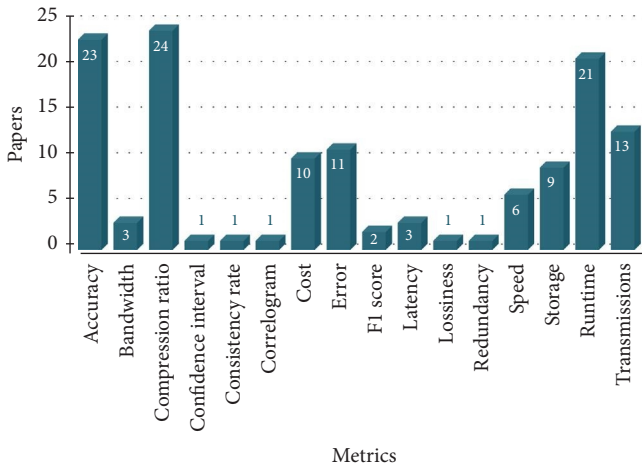


FIGURE 6: Use of performance metrics in reviewed articles.

one paper and the error rate in another. On average, at least three metrics were used in each paper, with the maximum being six and the minimum being two. It was observed that the number of metrics used in each paper is directly linked to the extent and/or depth of discussion of the results proposed by the authors. Subsection *Metrics* presents an in-depth analysis of metrics such as the objective of using a metric in different ways, the units of measurement, and the statistical methods used for the calculations.

The fourth question (RQ4: How are the proposed solutions compared?) refers to the method that the authors of each article used to compare and demonstrate their contributions and improvements concerning the state-of-the-art. From the classification mentioned in the previous question, it is possible to understand how the solutions are compared. It is a common sense that a solution must be compared with others of the same nature, an algorithm is compared with another algorithm, a technique with another technique, and so on. To understand the environment where the experiments were performed, the devices used, and the application layer, the data about the experiments were tabulated and are presented in Table 4. In this table, the information of the

experiments reported in the reviewed papers is summarized, through the columns: *Id/Ref*, the identifying number; *Solution Type*; *Layer*, to which the solution is addressed (E, edge; F, fog; C, cloud); *Test bed*, detailing the devices and/or environment used in the experiments; *Language*, used in code implementation; *Dataset*, informing the amount, form of access (PUB, public; PRI, private) and context; and *Compared to*, detailing how the solution is compared in the article.

Not all articles detailed the experiments: some cited, for example, a regular desktop as the experiment device; others did not specify the application layer or information about the data sets used, and some also did not give details about implementing the solution. Regarding the application layer, articles that did not specify the solution was specifically for the *edge* or *fog* layer were considered focusing a *cloud* as a computing platform. About the implementation language, for example, not all authors made it explicit, which justifies the missing values in the *Language* column. Regarding data publicity, most worked with publicly available data sets from traditional repositories (such as UCR time series classification archive [72] and UCI machine learning repository [73]), but some used private data, without making them available, due to corporate restrictions. Figure 7(b) presents the distribution of papers according the access policy of datasets.

An analysis of the data in Table 4 and the information in Figure 7 sought to identify the existence of a correlation between the type of solution and the layer and/or context of application of the solution. According to the study carried out, this correlation does not exist, which may indicate the need to identify the most appropriate techniques to be applied at each level. However, during the analysis carried out, some important observations can be made for the development of works in the area. The purpose of pointing out the layer for which the solutions are proposed, in the context of IoT, is because the requirements are different for each one, as detailed in section IoT.

Applications that require large amounts of data processing and storage, and high computational power, are more

TABLE 4: Summary information about experiment setups.

Id/Ref	Solution type	Layer	Test bed	Language	Data set	Compared to
A1 [32]	Data reduct. mechanism	E	Desktop	Python	1—PUB - weather	Papageorgiou's
A2 [33]	Data transmission control	E	Sensor + Arduino	Python	1—PRI - environment	Not compared
A3 [34]	Data filter module	F	Desktop	Python	2—PUB - weather	Not compared
A4 [35]	Dimension. red. approach	C*	Desktop Intel i7	Java	8—PUB - miscellaneous	PIP, TP
A5 [36]	TS compression scheme	C*	Desktop	Python	8—PUB - miscellaneous	GZIP, BZIP2, 7-zip
A6 [37]	Data reduction algorithm	C	Google Colab	—	1—PUB - Nasa Turbofan	Not compared
A7 [38]	TS classification algorithm	C	Intel Xeon	Java	50—PUB - miscellaneous	gRSE, INN + DTW, FS, LTS, SD, ELIS, BSPCover
A8 [39]	TS compression scheme	E	Sensor+Rasp. Pi	Python	5—BOTH - smart meter	ACS
A9 [40]	TS segmentation algorithm.	C*	Desktop	—	4—PVT - streaming	R function Smooth.spline
A10 [41]	TS lossless compression	EC	Sensors + Arduino	—	1—PRI - solar energy	Not compared
A11 [42]	TS compression algorithm	C*	Desktop Intel i7	Python	6—PUB - misc.	TRISTAN, Wavelet, Chebyshev
A12 [43]	ML-based data reduction	F	Macbook Pro i7	Matlab	2—PUB - weather	8 ML algorithms + RLE, Huffman
A13 [44]	TS compression algorithm	C*	Desktop Intel i7	—	1—PVT - solar energy	PCA, APCA, PWLH, SF, GHEB
A14 [45]	TS compression method	C*	Laptop Intel i7	Matlab	2—PUB - wearables	LTC, RLTC, ALTC
A15 [46]	ML data red. scheme	E	Sensor + Rasp. Pi	Python	4—PVT - weather	Shewhart
A16 [47]	Edge compr./Cloud forecast	EC	Laptop Intel i7	Python	1—PUB - traffic	HA, ARIMA, LSTM, GRU, ASTGCN
A17 [48]	Data transm. protocols	E	Desktop	C++	1—PUB - weather	PF, ATP-Harb
A18 [49]	Noisy TS autocoder	C	GPU Tesla K80	Python	1—PUB - mac. vibration	CNN (S, Res, DI, DS), Inception
A19 [50]	TS symbol approximation	EF	Sensor+Intel i3	—	1—PRI - accelerometer	Weasel Muse, RF, LR, 1-NN DTW
A20 [51]	TS representation framework	C	Cluster 300 Xeon	Matlab	128—PUB - misc.	Each step with similar approaches
A21 [52]	TS compression framework	C	Desktop	—	4—PVT - manufacturing	Byte Transfer, Huffman, Sprintz
A22 [53]	TS compressor	C	Intel Xeon + GPU	Python/C++	8—PUB - misc.	SZ, CA
A23 [54]	Hyper-sketching algorithm	EC	178 Xeon + Rasp.	Python	3—PUB - weather	LZ4
A24 [55]	TS Database	C	Intel Xeon	Java	7—PUB - misc.	InfluxDB
A25 [56]	Data compression method	C	Laptop Intel	C	6—PUB - miscellaneous	CA, SZ, LZ4
A26 [57]	TS analytics with comp.	C	Desktop Intel i7	C++	4—PUB - misc.	Sampling
A27 [58]	TS compression algorithm	C	Desktop Intel i5	C++	4—PUB - energy	ZSTD-22, Sprintz
A28 [59]	TS compression approach	E	Polar M600	C	12—PUB - wearables	CS, SZ, DWT
A29 [60]	TS compression algorithm	C	Macbook Pro i7	Python	89—PUB - misc.	SIMD, FastPFOR, LZ4...
A30 [61]	Set of PLA techniques	E	Edge device	—	3—PUB - traffic	SwingFilt., Convex-Hull, Contin., MixedPLA
A31 [62]	Data compression strategy	E	STM32L5 MCU	—	1—PUB - wind turbine	Not compared
A32 [63]	Embedded TS Database	C	Intel Xeon	—	2—PVT - misc.	InfluxDB, OpenTSDB
A33 [64]	ML-based compr. framework	EC	Desktop	—	2—PUB - miscellaneous	FDDL
A34 [65]	TS compression method	EC	Intel i7+ Rasp. Pi	C++17	7—PVT - misc.	Deflate, LZMA, Bzip2, LZ4, Snappy, Brotli, Zstd
A35 [66]	TS representation format	EF	Desktop	—	1—PUB - environment	JSON, STS
A36 [67]	Transfer learn. autoencoder	C	Desktop	—	3—PUB - biosignal	Stand Alone Model
A37 [68]	Chaching Scheme for TS	C	3 ThinkServers	C	5—PUB - misc.	Fatcache
A38 [69]	Data reduction approach	EF	OMNet++	C++	1—PUB - environment	PF, ATP, AVMDA, PIP-DA
A39 [70]	Data reduction algorithm	C*	Desktop	Python	1—PUB - misc.	Data PIP, AWBS, ApproxIoT, MaxMin
A40 [71]	Data reduction technique	E	Sensors + BeagleBone	C	1—PUB - maritime	ride, EdgeComp, slcc, Huffman, FastLZ, bzip2, cuSZ, LZ4

* Non-detailed information, assumed by the context.

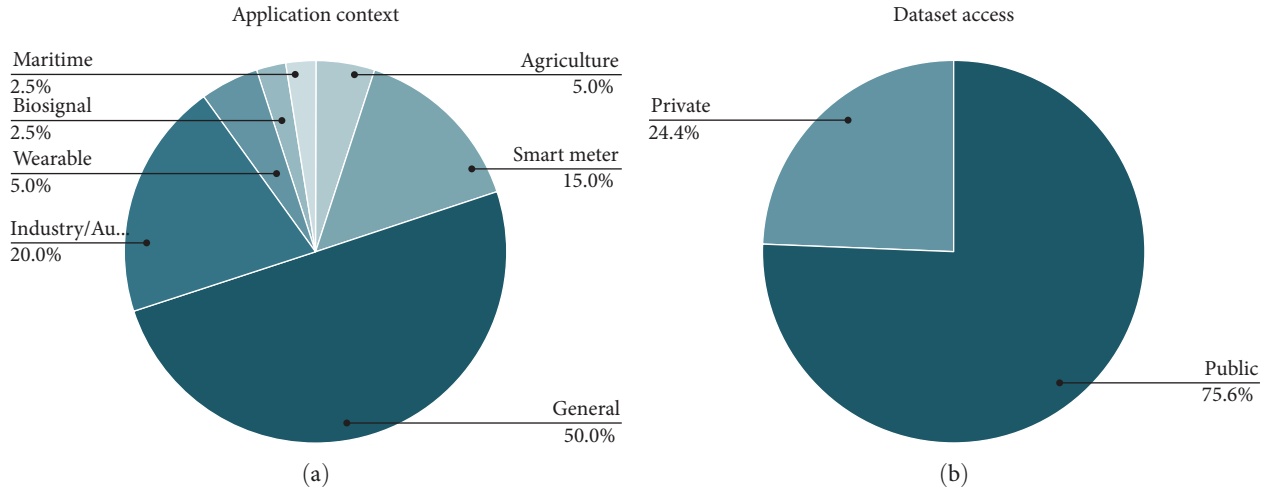


FIGURE 7: Distribution of papers according to the (a) context of the solutions and the (b) access policy of datasets.

suitable for cloud computing. The cloud-based IoT applications found in this review include large-scale analytics, ML, and artificial intelligence applications that process data from various sensors and devices. For example, papers A7, A20, and A37 perform experiments on servers or clusters with heavy computations. Also, in general, these are solutions that have been tested with many and large datasets, which is directly related to the need for data storage and confirms the importance of cloud architecture.

In other cases, fog computing is more suitable for IoT applications, especially when it requires near real-time data processing and high reliability. Fog computing enables processing of data near the edge of the network, closer to the source of the data, which reduces the latency and improves the response time. Examples of fog-based IoT applications include transmission control systems such as paper A3, data reduction at the gateway as paper A38, and monitoring systems, as papers A35 and A38, which handle industrial monitoring data.

Finally, edge computing is ideal for IoT applications that require real-time data processing, low latency, and low bandwidth. Edge-oriented solutions found in this review, in general, present experiments with edge devices, such as sensors, wearables and edge computers (such as Arduino and Raspberry), such as papers A2, A8, A15, A28, and A33. Edge computing processes data at the edge of the network, closer to the source of the data, which reduces bandwidth usage and improves response time.

In terms of numbers, there is a greater number of cloud solutions. Cloud technology is older than fog and edge, which may explain the greater number of cloud solutions. Cloud is also more consolidated and available in the technology market, especially in the Architecture as a Service (AaaS) model. Thus, it often becomes easier for researchers to buy a cloud service than to buy a sensor or edge computer and deploy it in IoT contexts. In particular, when the solution is aimed at contexts with peculiar characteristics, such as the industrial and agricultural environment. The development and dissemination of IoT technologies, in the present and

in the short-term future, tend to force development in the edge and fog layers, which, consequently, should boost research in these layers.

In response to the research question five (RQ5: Which of these solutions can be used in ML?), according to Table 2 (column *ML Alg.*), 14 papers were identified that use ML in one or more of the steps of the solution. The ML algorithms that appeared in the papers were k-NN, Random Forest and Deep Neural Networks such as CNN, RNN, LSTM, and autoencoders. In eight of those 14, the use of ML is in the compression process, while others six use the compression product as input for some classification or forecasting task.

The main idea of the eight papers that use ML in compression is in brief described as follows. In A2 [33] the authors propose the use of CNN to predict sensor readings, from a model trained with encoded time series, in order to explore spatio-temporal correlations in previous sensed data. In A3 [34], the k-NN algorithm is used to classify and filter the data. In A6 [37] the authors use an LSTM model to classify time series, using the model's feedback for future data reductions. In A15 [46], binary classification techniques help the application decide when to transmit data or not. In A18 [49] an autoencoder is proposed in order to extract meaningful information from raw time series. In A22 [53] the authors propose the use of neural networks for data prediction during the compression process. In A33 [64] authors proposes compression in IoT nodes and inference in base station or cloud, in both steps using DNNs, to reduce the volume of data transmitted in the network. And, in A36 [67] an autoencoder is also proposed, based on transfer learning, in order to perform data compression at the edge.

The other six papers use ML after compression. In A7 [38] the authors use approximation techniques for data reduction and then use these reduced data in time series classification, applying Random Forest. In A12 [43], first two compression algorithms are investigated, then they are combined with eight ML algorithms, to identify the best performance in an agricultural fog environment. In A19 [50] approximation techniques are also used, prior to the

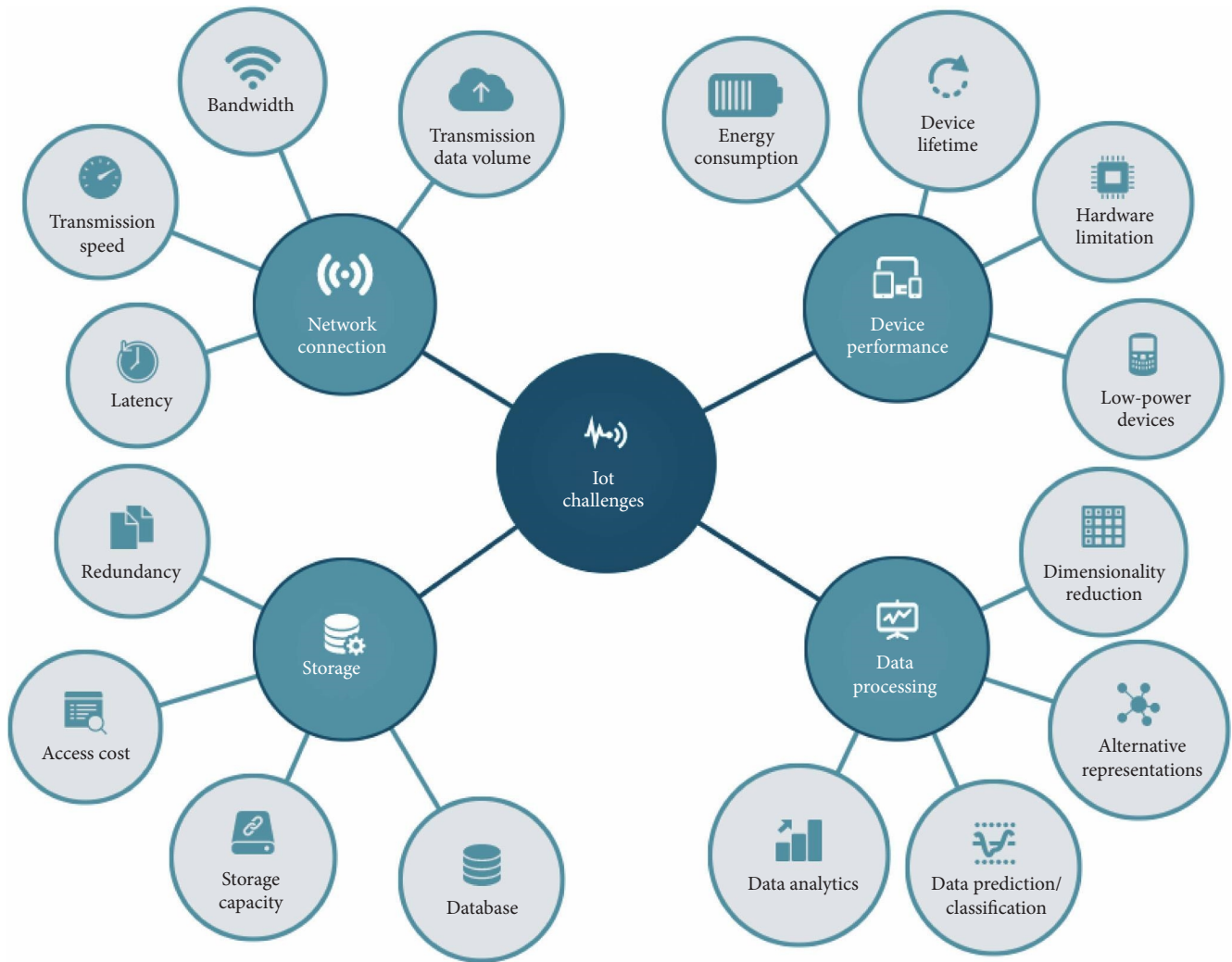


FIGURE 8: Major research challenges on time series compression for IoT applications.

application of ML, in order to reduce the volume of input data to the model. In A23 [54], the random forest algorithm was used in analytical tasks after compression. In A28 [59], recurrent and convolutional neural networks are used to classify time series after compression. And, in A39 [70], a long short-term memory (LSTM), with Adam optimizer and binary cross entropy as a loss function, was used to classify time series.

Here is an interesting point about using ML during or after compression. Fundamentally, edge devices have limited capabilities compared to desktop computers or servers. However, given the recent technological evolution, it is possible to observe that IoT devices are already able to deal with the computation of ML models. This expands the horizons of IoT applications, which go beyond the simple data collection. A deeper discussion of using ML before or after compression is provided in the subsection *Time Series Compression and Machine Learning*.

Regarding question six, the aspects that motivated the development of solutions for time series compression in IoT applications (RQ6: What factors motivate the use of time series compression solutions in IoT applications?).

They are also listed in Table 2, in the *Goal* column. Figure 8 presents an infographic that helps the understanding of the challenges listed by the authors of the selected papers, which served as motivation for the development of solutions. Some solutions have over one powerful motivation, but it is observed that the solutions seek to reduce the consumption of resources in IoT environments. Among the motivations presented, the following are highlighted: reduction in the need for storage, given the hardware limitations of the devices; the reduction in data transmission, given the scarcity and cost of communication in many IoT environments; the reduction in energy consumption, to provide the necessary autonomy to the devices; and optimize the way of storing and indexing large volumes of time series and, consequently, reducing the cost of accessing this data.

Another way to map the motivating factors of the works is from the keywords. Figure 9 presents a graph, composed of the keywords of the 40 papers, demonstrating which are the most used and the relationships with sub keywords. This graph was also generated with VOSviewer tool. From the analysis of the graph, it is possible to observe, in addition to the main terms, the subgroups of interrelated problems,

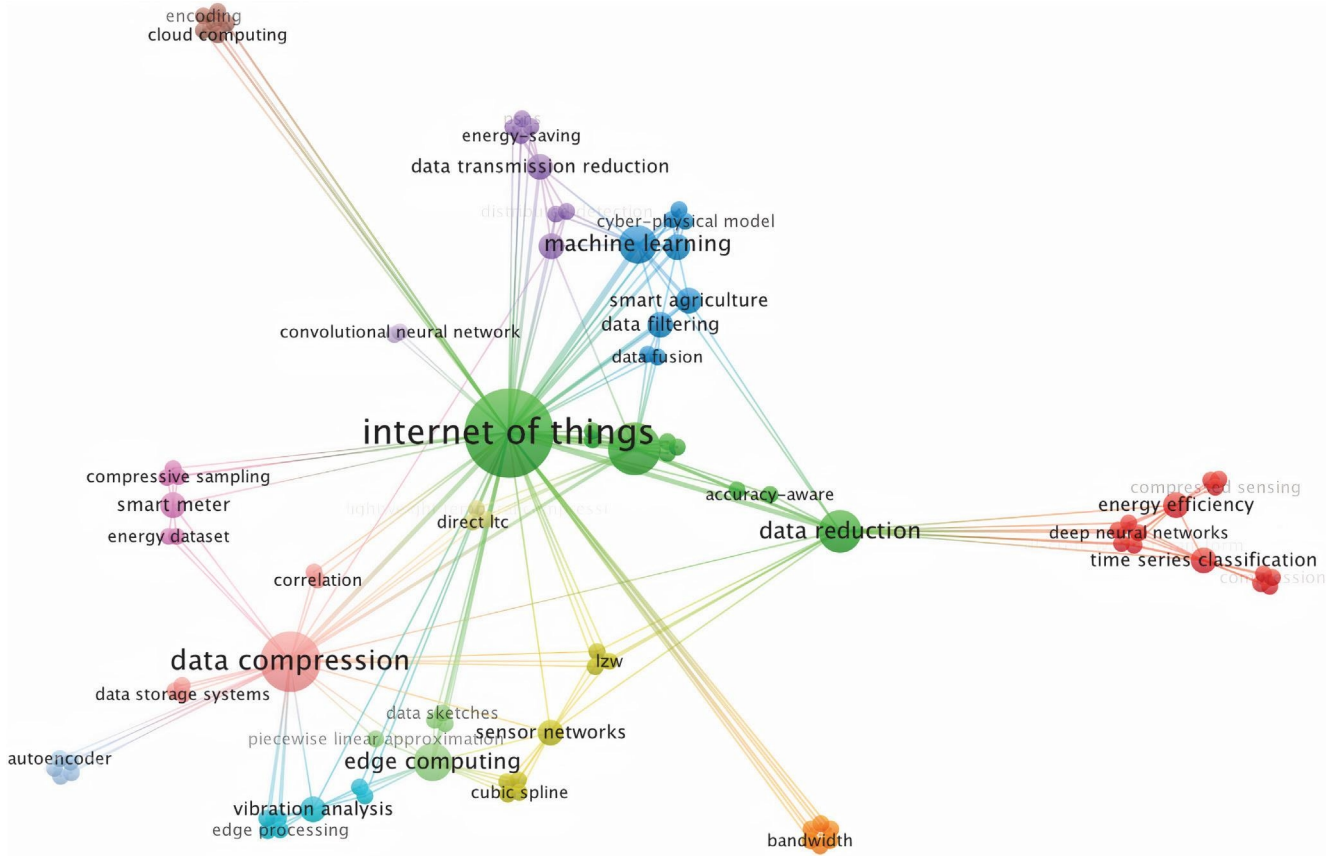


FIGURE 9: Graph with links between the keywords of the 40 selected papers.

such as ML and smart agriculture, data compression and data storage systems, data transmission reduction and energy saving, and so on.

To answer question seven (RQ7: In what context were these solutions developed?), Table 2 also presents this information in the *Context* column, and Figure 7(a) illustrates the distribution by context. Most solutions apply to the time series in any context stated in the table as *General*. However, some solutions are context-oriented, given the peculiarity of applications, such as agriculture, industry, and wearables.

In response to the eighth question (RQ8: Which of these solutions were developed in agriculture?), only three techniques developed in agriculture were found (*A1* and *A12*, from same authors, and *A3*). Although most of the solutions reviewed are developed with a general purpose, the agricultural context, which is the focus of other work fronts for researchers, has very peculiar characteristics, especially concerning the availability of resources. In all these three papers, the main motivation is the reduction in data transmission, corroborating the lack of abundant connectivity as a significant difficulty of agricultural IoT applications. Thus, it is interesting to investigate what solutions were developed in this context and, consequently, to highlight the existence of a few alternatives for this environment, a research gap.

In general, all these research questions help us to understand the scenario related to time series compression and serve as a basis for future advances in the state of the art.

Moreover, given the need for an adequate comparison between existing solutions, it is equally important to analyze the metrics used by the authors to validate of the solutions presented in the reviewed papers. Thus, the next subsection deepens this analysis, to summarize and score relationships between the metrics.

4.3. Metrics. To understand the solutions presented in the reviewed papers, besides understanding how they work, it is necessary to detail some aspects of the metrics used in performance measurement. The performance of time series compression solutions can be measured from several perspectives [74]. It is possible to analyze, for example, the complexity of the algorithm or technique, the computational cost involved, the speed of operation, the compression rate, or the quality of the reconstructed data. Based on an in-depth analysis, information about all metrics that appeared in any paper selected for this review was summarized in Table 5. This table presents, for each metric used (column *Metric*), the aspect for which this metric was used (column *Regarding*), how the authors calculated the reported value (column *Calculation*), as well as the unit of measurement applied to each situation (column *Unit of Measurement*). Each row in the table represents an occurrence, in at least one paper, of the metric in the first column related to the task in the second column. Some metrics appear more than once, with more than one way to calculate and expressed in different units of

TABLE 5: Overview of the metrics and coefficients used in the reviewed papers.

Metric	Regarding	Calculation	Unit of measurement
Accuracy	Classification	Hits, AUC score	
	Fusion	MSE	
	Prediction	Hits, RMSE, MAPE, AUC score	%
	Recovery	Jaccard, MSE, RMSE, NRMSE, ISD, PRD	
	Regression	RMSE	
Bandwidth	Transmission	Lost data	
	Consumption	Data sent	Bytes
Compression ratio	Saving	Specific equation	%
	Compression	Specific equation, Nemenyi test, data reduction	Bytes, %
Confidence interval	Certainty	Specific equation	%
Consistency ratio	Cloud/nodes	RMSE	Percentage
Correlogram	Correlation	Density and dispersion plots	Graphic
Cost	Battery level	Device information	Graphic
	Energy consumption	Specific equation	Joules
	Randon access	Bits accessed	Bits
	Sampling space	Chernoff bounds	%
Error	Compression	RMSE, loss	%
	Guarantee	Specific equation	$[-1,1]$
	Quantization/noise	Specific equation, SNR	%
	Reconstruction	Normalized RMSE	%
	Relative/true error	Specific equation	%
F1 score	Classification	Specific equation	%
Latency	Compression	N of records added	Records
	DB query	Time	s
Lossiness	Compression	RMSE	%
Redundancy	Compression	N of redundant pairs	Redundant sets
Runtime	Compression	Time	
	Execution	Time	
	ML training	Time	Milliseconds
	Reconstruction	Time	
	Index construction	Time	
	MR representation	Time	
Speed	Comp. + Decomp.	Specific equation	
	Compression	Specific equation	MBps
	Decompression	Specific equation	
	Forecaster	Specific equation	
Storage	Consumption	Date size difference	Bytes
	Data size	Date size difference	Bytes, %, IOPS
	Space saving	Specific equation	%
Transmissions	Data	Rate of transmissions	Bytes, %, sent packets
	Payload	Size	Bytes
	Throughput	Speed, OPS	MBps, OPSps

measurement. The number of occurrences of each metric is shown in Figure 6.

The purpose of constructing the table is to deepen the analysis of metrics, especially concerning the statistical method used to calculate the values presented by the authors. Although the focus of the works is the same, the specific contributions of each work differ in nature and objectives.

Therefore, it is necessary to identify the purpose of use of each metric, to enable a fair comparison of the state-of-the-art applications. Moreover, an important issue to note is that metrics are not used in a single and isolated way, but together and related in a specific context. Thus, it is also common for the same index, such as the *Root-Mean-Square Error (RMSE)*, to be interpreted from one perspective as a metric and from

another perspective as another metric: *Accuracy* or *Error*, for example. Even the statistical methods relate complementary metrics: for example, *Space Saving* can be calculated as $1 - \text{Compression Ratio}$.

Regarding the three most used metrics among the reviewed papers (*Accuracy*, *Compression Ratio*, and *Runtime*), it was observed that the form of measurement is intrinsically related to the task to be evaluated. *Accuracy* concerns the level of correctness in performing a task, so, in data classification, this metric is related to the number of right classifications, while in data transmission it can refer to the number of data packets correctly transmitted. The *Compression Ratio* is directly related to the efficiency gain from compressing a data set, and this gain can be represented by the space saved in bytes or by the percentage reduction in the final size of the data. And the *Runtime* is the difference between the start and end time in the execution of a task, counted in the computation of CPU time.

4.4. Threats to Validity. This work, while a SLR, is subject to threats to its validity, like any other scientific work. The fundamental aspect of concern is concerning study selection bias, both in the search and in the filtering of studies. First, regarding the search process, were used the most relevant terms and keywords, as well as the main databases in the area. However, some relevant work still may not have been returned by the search, as there is no intersection between its title, abstract terms, or keywords and the searched ones. Undetected problems in database search engines may also have affected this process. To mitigate this possibility, an exhaustive search string calibration phase was performed and reviewed by all authors more than once, before conducting the review, as presented in Section Research Methodology, Subsection Planning. In addition, the final search was repeated by all authors, and the results were subsequently consolidated to increase the robustness of the process. Related to the filtering of studies, some exclusion criteria were applicable (data, short papers, and language), and others required reading elements such as titles and abstracts. Therefore, the selection or not of a paper may vary according to the authors' understanding of the work. To increase the robustness and reliability of the filtering process, all reviews performed by the authors were peer-reviewed, as detailed in the research methodology (Section Research Methodology).

5. Review Findings and Discussions

This section presents a discussion of the results of the SLR described in this paper, as well as future research directions identified by the authors. These discussions start with the analysis of the data and are based on the knowledge inferred after the synthesis of the information. In addition, it also provides some potential trends to discuss relevant research gaps that could be directed to other research opportunities.

5.1. Time Series Compression and Machine Learning. The major aim of deploying an IoT system is to automate a data collection procedure. These data serve as input for further processing, where artificial intelligence techniques often appear, especially those under ML. Here, an important

problem arises: to increase the quality of models trained with ML, the maximum available data are used. Only the quantity does not guarantee success in training a model, it is necessary to observe the aspect of data quality.

The relationship between time series, compression, and ML can exist in many ways. In a macro view, it is possible to reflect on this by separating between using ML *for the compression*, to improve the process of compressing and/or decompressing data, or *after compression*, using the compressed data as input for tasks, such as time series classification or prediction. Therefore, the use of ML in this process is still little explored and this is the main research gap identified by this review. As shown in Table 3, only 14 of the 40 papers reviewed explore the use of ML algorithms, and not all of them do so for the purpose presented above, as cited in section Results, subsection Answers to the research questions, when answering RQ4.

In the first case, there are some compression techniques today based on ML, mainly those based on the neural networks, with emphasis on autoencoders. However, such algorithms have considerable computational costs involved in model training. Within the context of the proposed work, which focus on edge computing and IoT devices, computational costs are an important decision factor for adopting a technique or strategy. On the other hand, the second case, using compressed data as input for ML, is much more common and widely applied in other scopes, besides IoT. Since the concept of IoT emerged until today, the significant majority of IoT applications are focused on collecting data to serve as input for building ML models. As the storage and transmission of data is a perennial problem in computing, over time compression techniques have been applied to reduce the volume of data, for later use of these compressed data in ML modeling.

Since the central purpose of applying IoT systems is data collection, and the main input of ML algorithms is the data collected by IoT sensors, this research gap deserves a lot of attention. From this, solutions that can compress data collected by IoT devices without causing a loss of quality have a great potential for technological impact. Regardless of the purpose of data collection, both for time series classification or prediction tasks, there is room for improvement and advances in exploring time series compression and ML algorithms in a complementary way.

Furthermore, analyzing the possibilities within the scope of this work, from the perspective of IoT devices, and given the technological evolution of recent years, it is quite possible and likely that within a short time the computational power of edge devices, which is already quite impressive, will reach an even greater level, never seen or thought of in computing. This will make edge devices better handle heavy computational tasks, especially in terms of processing and storage costs, such as training models with many ML techniques. This will certainly expand the possibilities and encourage the use of ML, not only consuming compressed data, but also for data compression in IoT applications.

5.2. Compression Algorithms Design. Data compression is a critical issue in IoT domain, where devices generate massive

amounts of data. In recent years, significant advances have been made in data compression techniques, driven by advances in technology. The primary methods for data compression in IoT are statistical methods, both lossless and lossy. The use of lossless compression techniques, such as papers A5, A10, A21, and A25, started with Huffman and arithmetic coding, has been prevalent in IoT applications over the past decade. Other papers, such as A3, A17, A30, and A38 use lossy compression methods to filter the data, removing noise, or other unnecessary information. These techniques compress data by exploiting the statistical properties of the data and encoding it more efficiently. This approach is well-known and applicable in several situations; however, it has limitations in terms of performance, since the more complex the computation, the heavier the task becomes, and, therefore, it can impair the performance of an IoT application with limited resource. Statistical methods for data compression in IoT can be useful in some contexts, particularly when the data is less structured and contain less predictable patterns. In addition, statistical methods may be less vulnerable to adversarial attacks, making them a more secure option in some cases, and these methods do not require large amounts of labeled training data, making them a more feasible option in situations where labeled data is not available.

Another significant development in data compression for IoT is the use of ML algorithms, which can be used to compress data more efficiently by learning the patterns in the data and compressing it accordingly, such as papers A3, A12, A15, and A19. In addition, deep learning algorithms such as autoencoders, as used in papers A18, A33, and A36 can be used to perform unsupervised data compression, where compressed representation of the data can be used for downstream tasks such as anomaly detection. Looking ahead, the possibilities of data compression in IoT is very possible to be driven by advancements in ML techniques. The use of generative models such as generative adversarial networks (GANs) and variational autoencoders (VAEs) for data compression is an exciting area of research that has shown promising results. Furthermore, the use of federated learning, where models are trained locally on IoT devices and then aggregated on a central server, can enable efficient compression of IoT data while preserving privacy. Using ML for data compression in IoT has several advantages, such as better compression, real-time compression, robustness, and scalability. However, it also has weaknesses, such as high computational requirements, training data requirements, interpretability, and security concerns. In summary, when deciding whether to use ML for data compression in IoT, it is important to consider mainly aspects such as the data characteristics, security requirements, and training data availability. Careful consideration of these factors can help determine if ML is the appropriate approach for a particular IoT context.

It is hard to conclude with certainty which design for compression algorithms in IoT is most likely to be used in the future, as both ML and statistical methods have their strengths and weaknesses depending on the specific context and application. The choice between ML, statistical methods, or even other methods for data compression in IoT will depend on the specific context and requirements of the

IoT application. A deep comparison between solutions in terms of algorithm design was not possible with the elements provided by the authors of the reviewed papers. Some authors did not publicize the scientific artifacts produced or even the information reported in the paper is insufficient for its replication. Also, the wide variation in the focus, structure and nomenclature of the solutions makes the task of comparison complex. This work makes an initial contribution in this regard, grouping relevant and interesting information at first, but also highlights an opportunity to deepen the development of scientific artifacts such as a taxonomy, in order to better group similar solutions, with a similar focus and structure.

5.3. Main Outcomes and Challenges. The main findings and challenges from the review are described as follows. As exemplified in Figure 8, the main current challenges related to time series compression refer to the IoT context. The limitation of resources such as hardware, connectivity, computational power, and energy efficiency is inherent to the IoT concept and will remain as a major challenge in the coming years, even with technological advances. This is because in the same proportion that the devices and communication conditions improve, the complexity of the solutions and the demand for IoT systems also increase. This circumstance makes the solutions permeate the frontier of knowledge, updating the state-of-the-art at a high frequency.

In addition, the spread of IoT devices has made it possible to explore these systems in the most diverse environments and will continue to expand the possibilities even further. With the increase in demand for IoT devices and the consequent strengthening of the production chain, new manufacturers are entering the market with new device models, which increase the potential for new applications, while lowering the cost of older or simpler devices. Thus, it is essential that researchers monitor the market to follow the evolution of devices and take advantage of the technological timing, especially with the opening of new digital possibilities, which are large because of the technological impact of the COVID-19 pandemic, on the present and future.

The results were analyzed from different perspectives, especially considering the type of solution proposed, the devices and the setup of the experiments presented, the evaluation metrics used, the use of ML techniques in the proposals, and the importance of the IoT context for which the solution was used is addressed.

5.3.1. Solutions Type. The time series compression problem is addressed in several ways. In this review, the solutions were classified into algorithms, techniques, extensions, and combinations. This classification provides a greater view and transparency in the explanation of results and the comparison between solutions. It was observed, for example, that algorithms are analyzed in terms of execution time and computational cost, while frameworks and tools are more concerned with the execution flow of the application and with the capacity of the solution to use the data.

It is noteworthy that researchers are aware of the particular nature of solutions and that the authors make these

aspects clear through the nomenclature of solutions. There are many universal terms used to name the solutions, such as schema, mechanism, approach, and method. These terms are still used for applications of different natures. Therefore, the need to develop a taxonomy stands to organize existing solutions and guide future solutions. In this review, an incipient classification of the solutions presented in the reviewed papers is proposed, which can be enhanced and complemented to cover the current categories and definitions.

5.3.2. Devices and Experiments. Conducting experiments is a fundamental part of all scientific work. However, these performance tests must be performed in a trustworthy context in which the solution is designed. Among the papers reviewed, low use of IoT devices was found in the tests, especially in those solutions targeted at the edge layer. Given the significant differences in hardware between simulators running on desktop and real-IoT devices, the authors must take greater care in evaluating the proposed solutions, especially in choosing the devices for their test beds. The cost of most IoT devices is reduced, given their dissemination and the consequent wide market demand. This makes a wide range of devices accessible to researchers, which could test solutions in a scenario closer to the real world.

In addition, a minor concern of the authors with the complete description of the experiments performed was identified. Many papers do not present details of the devices used, their specifications, the development language used in the implementation, or clarity in the explanation of the algorithm, as can be seen in Table 4. This information must be perfectly publicized for greater reliability of the proposed solutions and to permit faithful comparisons between distinct approaches.

5.3.3. Evaluation Metrics and Datasets. Considering the evaluation metrics used by the authors, the effect of using the main metrics linked to data compression became clear. The three metrics, *compression ratio*, *accuracy*, and *runtime* represent the variables that demand to be adjusted in the compression equation: computational cost, quality of the compressed data, and application response time. The papers that did not bring evaluation of the prospect of these three metrics were those that moved a little away from solving the problem of data compression, concentrating on complications perpendicular to volume reduction, such as energy efficiency and communication.

The importance of using public and well-known datasets for experimenting with solutions was also highlighted. Most authors carried out experiments and evaluations using real and public data, or to collect new data and make them available. Authors who use private datasets make it impossible to replicate the studies and have their achieved performance audited and validated by a third party. Considering that there are today several public repositories with a large volume and variety of data, it is important to show the performance of a solution through this data.

5.3.4. Solutions Context. Regarding the context of the solutions, it was observed that the vast majority did not address the solution to a specific time series, being classified for

reporting purposes, as a general context. Although there are intersections between different environments where IoT applications are used. This result shows that there is a wide variety of contexts where solutions for time series compression have not yet been explored. There may be many reasons for this lack of context specification, from the unavailability of an environment for testing solutions, the relative incipience in the application of IoT systems, or even the lack of resources for the implementation of IoT systems in some contexts. In the agricultural context, for example, which is one of the interests of the research group to which the authors of this work belong, even though there are many efforts in the implementation of agricultural IoT systems, they are still scarce because of producers' resistance to technology or due to requiring a high-financial investment [75].

However, the particularities of the context can not be ignored by researchers, at the risk of making the application of solutions in real scenarios unfeasible. It is necessary to realize the importance of the specificity of the context, limitations, and characteristics of each environment, to develop solutions that apply and with high potential for impact on environments [76]. Thus, the results obtained will be closer to generating economic gains in real scenarios.

5.4. Future Research Directions. This review summarizes several results adopted to address the challenges in time series compression for IoT. Although some solutions present interesting results concerning the state-of-the-art, there is still a need to address many open challenges.

Overall, from the analysis performed, it emerged that: (i) most of the reviewed papers deal with time series compression *techniques* (technique, extension, or combination), using over one algorithm, while few authors propose new algorithms; (ii) the use of time series compression techniques for IoT combined with ML, so far, has been little explored, especially from the perspective of analyzing the effect of compression on the development of smart models; (iii) few papers consider the particularities of the contexts where IoT solutions are applied, which have quite different restrictions in each environment; (iv) very few authors have tested their solutions in actual situations, most tested their solutions using data gained in controlled or simulated environments.

These open issues that came up from the reviewed papers can be summed up in the following categories:

Explore ML: given that the main objective of using IoT systems is to collect data for artificial intelligence applications, the finding of only two related data compression with ML techniques highlights a significant research gap and the need for further investigation into alternatives for this purpose.

Perform real experiments on real devices: provided the difference in resource availability between a computationally emulated environment and a proper environment where IoT devices are used, it is essential to carry out experiments in real scenarios to consider the performance of the solutions. Furthermore, with the recent technological evolution and the consequent reduction

in the cost of IoT devices, nowadays, most of these pieces of equipment are accessible to researchers at a low investment.

Explore edge computing: most of the solutions reviewed are focused on the cloud layer, they do not explore distributed computing in the fog and edge layers, centralizing the processing. Additionally, because of technological advances in recent years, today, the computing power of IoT devices is much higher and is growing, which makes it possible to explore processing, especially at the edge.

Investigate the particularities of the different IoT contexts: although there are intersections in technical aspects between IoT solutions applied to different contexts, there can be big differences concerning the availability of resources in other IoT environments. Researchers must examine these particularities, as well as the time series in focus, when developing new solutions to ensure proper application performance.

6. Conclusion

In this paper, we have performed a descriptive systematic review of time series compression for IoT, following the protocol presented by Kitchenham and Charters [26]. First, the review methodology was described. The main tool used was *Parsifal*, through which the stages of planning, conducting, and reporting were carried out. Some artifacts were presented in deep, such as the protocol, the quality assessment checklist, and the data extraction form, as well as information about the papers selected for review. From 1,812 papers in the initial search, 40 were selected for review, based on the inclusion and exclusion criteria and their relevance to the review.

After selecting the papers for review, the solutions proposed were summarized through comparative tables and images, and these artifacts were used to answer the research questions initially defined. Information about the experiments performed, such as setup, test bed, devices, and implementation was also reported. An in-depth discussion about metrics was also presented, dealing with which metrics were used and in which cases, in addition to a report on the statistical and mathematical methods used to calculate them and the units of measurement used. As the main finding of this review, the research gap regarding the use of ML and time series compression deserves to be highlighted. Then, the main outcomes of the study were described and highlighted by the findings of the review, emphasizing the relationship between time series compression for IoT application and ML, and splitting the use of ML during and after compression. The review showed the importance of the topic for the evolution of applications in IoT, clarifying that recent technological developments enable the development of a wide variety of solutions for focused smart environment applications. Finally, it were highlighted that some issues are still open to the topic, which proves that further research efforts are required to develop the process of compression series format in the IoT context.

Data Availability

No datasets were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES), Finance Code 001.

References

- [1] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, and A. Hussain, "Big data and IoT-based applications in smart environments: a systematic review," *Computer Science Review*, vol. 39, Article ID 100318, 2021.
- [2] S. K. Jensen, T. B. Pedersen, and C. Thomsen, "Time series management systems: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2581–2600, 2017.
- [3] M. Bansal, I. Chana, and S. Clarke, "A survey on iot big data: current status, 13 V's challenges, and future directions," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–59, 2020.
- [4] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: a survey of problems and methods," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–41, 2018.
- [5] U. Jayasankar, V. Thirumal, and D. Ponnurangam, "A survey on data compression techniques: from the perspective of data quality, coding schemes, data type and applications," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 2, pp. 119–140, 2021.
- [6] G. Chiarot and C. Silvestri, "Time series compression: a survey," 2021.
- [7] L. Wen, K. Zhou, S. Yang, and L. Li, "Compression of smart meter big data: a survey," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 59–69, 2018.
- [8] F. Wortmann and K. Flüchter, "Internet of things," *Business & Information Systems Engineering*, vol. 57, pp. 221–224, 2015.
- [9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [10] P. Friess, P. Guillemin, H. Sundmaeker, European Commission, and Directorate-General for the Information Society and Media, *Vision and Challenges for Realising the Internet of Things*, Publications Office of the European Union, Luxembourg, 2010.
- [11] F. Wang, L. Hu, J. Zhou, and K. Zhao, "A survey from the perspective of evolutionary process in the internet of things," *International Journal of Distributed Sensor Networks*, vol. 11, no. 3, 2015.
- [12] I. Bojanova, J. Zhang, and J. Voas, "Cloud computing," *IT Professional*, vol. 15, no. 2, pp. 12–14, 2013.
- [13] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011, Computer Security Division - National Institute of Standards and Technology <https://csrc.nist.gov/publications/detail/sp/800-145/final>.
- [14] M. Hajibaba and S. Gorgin, "A review on modern distributed computing paradigms: cloud computing, jungle computing and fog computing," *Journal of Computing and Information Technology*, vol. 22, no. 2, pp. 69–84, 2014.

- [15] A. Jain and P. Singhal, "Fog computing: driving force behind the emergence of edge computing," in *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pp. 294–297, IEEE, 2016.
- [16] F. Al-Doghman, Z. Chaczko, A. R. Ajayan, and R. Klempous, "A review on Fog Computing technology," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 001525–001530, IEEE, 2016.
- [17] A. Ahmed, H. R. Arkian, D. Battulga et al., "Fog computing applications: taxonomy and requirements," 2019.
- [18] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: a review," *Big Data and Cognitive Computing*, vol. 2, no. 2, Article ID 10, 2018.
- [19] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [20] D. Salomon, *Data Compression: The Complete Reference*, Springer, London, 4th edition, 2007.
- [21] H. M. Cooper, "Organizing knowledge syntheses: a taxonomy of literature reviews," *Knowledge in Society*, vol. 1, Article ID 104, 1988.
- [22] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasar, "How-to conduct a systematic literature review: a quick guide for computer science research," *MethodsX*, vol. 9, Article ID 101895, 2022.
- [23] G. Paré, M.-C. Trudel, M. Jaana, and S. Kitsiou, "Synthesizing information systems knowledge: a typology of literature reviews," *Information & Management*, vol. 52, no. 2, pp. 183–199, 2015.
- [24] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, Chapman and Hall/CRC, New Yor, 2015.
- [25] A. Osorio, "A guidelines proposal for computer science research (in portuguese)," Master's thesis, Federal University of Pelotas, Pelotas, RS, Brazil, 2020.
- [26] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University Joint Report, 2007, Technical report https://www.elsevier.com/___data/promis_misc/525444systematicreviewsguide.pdf.
- [27] C. Okoli, "A guide to conducting a standalone systematic literature review," *Communications of the Association for Information Systems*, vol. 37, pp. 879–910, 2015.
- [28] M. Oliveira Jr., G. Sedrez, G. de Souza, and G. G. H. Cavalheiro, "An application with jetson nano for plant stress detection and on-field spray decision," in *Proceedings of the 11th International Conference on Sensor Networks - SENSORNETS*, pp. 215–222, SciTePress, 2022.
- [29] M. A. Oliveira Jr., G. Sedrez, A. Monteiro, F. E. Puntel, and G. G. H. Cavalheiro, "Effects of agro-sensor time series approximation on plant stress detection: an experimental study," in *Proceedings of the 13th Congresso Brasileiro de Agroinformática*, pp. 99–107, Sociedade Brasileira de Computação, Porto Alegre, RS, Brasil, 2021.
- [30] RNP, "Comunidade acadêmica federada - CAFe," 2021, <https://www.rnp.br/servicos/cafe>.
- [31] CAPES, "Portal de periódicos," 2021, <http://www.periodicos.capes.gov.br/>.
- [32] L. Feng, P. Kortoçi, and Y. Liu, "A multi-tier data reduction mechanism for IoT sensors," in *Proceedings of the Seventh International Conference on the Internet of Things*, pp. 1–8, Association for Computing Machinery, New York, NY, USA, 2017.
- [33] J. Han, G. H. Lee, S. Park, J. Lee, and J. K. Choi, "A multivariate-time-series-prediction-based adaptive data transmission period control algorithm for IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 419–436, 2022.
- [34] F. M. Ribeiro, R. Prati, R. Bianchi, and C. Kamienski, "A nearest neighbors based data filter for fog computing in IoT smart agriculture," in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, pp. 63–67, IEEE, 2020.
- [35] Y. Hu, Z. Jiang, P. Zhan, Q. Zhang, Y. Ding, and X. Li, "A novel multi-resolution representation for streaming time series," *Procedia Computer Science*, vol. 129, pp. 178–184, 2018.
- [36] R. Vestergaard, D. E. Lucani, and Q. Zhang, "A randomly accessible lossless compression scheme for time-series data," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pp. 2145–2154, IEEE, 2020.
- [37] T. Hafeez and G. McArdle, "Accuracy-aware data reduction for internet of things," in *2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, pp. 57–62, IEEE, 2022.
- [38] J. Yang, S. Jing, and G. Huang, "Accurate and fast time series classification based on compressed random shapelet forest," *Applied Intelligence*, vol. 53, pp. 5240–5258, 2023.
- [39] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive multivariate data compression in smart metering internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1287–1297, 2021.
- [40] R. Dębski and R. Drezewski, "Adaptive segmentation of streaming sensor data on edge devices," *Sensors*, vol. 21, no. 20, Article ID 6884, 2021.
- [41] H. M. El-Hageen, H. Albalawi, A. M. Alatwi, W. R. Abd Elrahman, and S. T. Mohammed Faqeh, "Agile lossless compression algorithm for big data of solar energy harvesting wireless sensor network," *Sensors and Materials*, vol. 34, no. 11, pp. 4095–4111, 2022.
- [42] A. Khelifati, M. Khayati, and P. Cudré-Mauroux, "CORAD: correlation-aware compression of massive time series using sparse dictionary coding," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2289–2298, IEEE, 2019.
- [43] F. M. Ribeiro Junior, R. A. C. Bianchi, R. C. Prati, K. Kolehmainen, J.-P. Soininen, and C. A. Kamienski, "Data reduction based on machine learning algorithms for fog computing in iot smart agriculture," *Biosystems Engineering*, vol. 223, Part B, pp. 142–158, 2022.
- [44] B. Ruan, W. Hua, R. Zhang, and X. Zhou, "Decomposition-based approximation of time series data with max-error guarantees," in *Databases Theory and Applications*, pp. 71–82, Springer, 2017.
- [45] L. Klus, R. Klus, E. S. Lohan et al., "Direct lightweight temporal compression for wearable sensor data," *IEEE Sensors Letters*, vol. 5, no. 2, pp. 1–4, 2021.
- [46] A. P. Singh and S. Chaudhari, "Embedded machine learning-based data reduction in application-specific constrained IoT networks," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 747–753, Association for Computing Machinery, New York, NY, USA, 2020.
- [47] X. Du, Z. Zhou, Y. Zhang, and T. Rahman, "Energy-efficient sensory data gathering based on compressed sensing in IoT networks," *Journal of Cloud Computing*, vol. 9, Article ID 19, 2020.
- [48] A. K. Idrees, R. Alhussaini, and M. A. Salman, "Energy-efficient two-layer data transmission reduction protocol in periodic sensor networks of IoTs," *Personal and Ubiquitous Computing*, vol. 27, pp. 139–158, 2023.

- [49] M.-A. Tnani, P. Subarnaduti, and K. Diepold, "Extract, compress and encode: LitNet an efficient autoencoder for noisy time-series data," in *2022 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1–8, IEEE, 2022.
- [50] D.-K. Choi, "Fog computing application of cyber-physical models of IoT devices with symbolic approximation algorithms," *Journal of Cloud Computing*, vol. 11, Article ID 63, 2022.
- [51] J. Paparrizos and M. J. Franklin, "GRAIL: efficient time-series representation learning," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1762–1777, 2019.
- [52] Q.-T. Doan, A. S. M. Kayes, W. Rahayu, and K. Nguyen, "Integration of IoT streaming data with efficient indexing and storage optimization," *IEEE Access*, vol. 8, pp. 47456–47467, 2020.
- [53] S. Chandak, K. Tatwawadi, C. Wen, L. Wang, J. A. Ojea, and T. Weissman, "LFZip: lossy compression of multivariate floating-point time series data via improved prediction," in *2020 Data Compression Conference (DCC)*, pp. 342–351, IEEE, 2020.
- [54] T. Buddhika, M. Malensek, S. Pallickara, and S. L. Pallickara, "Living on the edge: data transmission, storage, and analytics in continuous sensing environments," *ACM Transactions on Internet of Things*, vol. 2, no. 3, pp. 1–31, 2021.
- [55] N. Agrawal and A. Vulimiri, "Low-latency analytics on colossal data streams with SummaryStore," in *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 647–664, Association for Computing Machinery, New York, NY, USA, 2017.
- [56] V. Agrawal, G. Kuldeep, and D. Dey, "Near lossless time series data compression methods using statistics and deviation," 2022.
- [57] C. Lin, E. Boursier, and Y. Papakonstantinou, "Plato: approximate analytics over compressed time series with tight deterministic error guarantees," *Proceedings of the VLDB Endowment*, vol. 13, no. 7, pp. 1105–1118, 2020.
- [58] J. R. Ward and S. K. Barker, "Powerstrip: high-performance compression for energy data," in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pp. 242–252, Association for Computing Machinery, New York, NY, USA, 2020.
- [59] J. Azar, A. Makhoul, R. Couturier, and J. Demerjian, "Robust IoT time series classification with data compression and deep learning," *Neurocomputing*, vol. 398, pp. 222–234, 2020.
- [60] D. Blalock, S. Madden, and J. Guttag, "Sprintz: time series compression for the internet of things," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–23, 2018.
- [61] R. Duvignau, V. Gulisano, M. Papatriantafyllou, and V. Savic, "Streaming piecewise linear approximation for efficient data management in edge computing," pp. 593–596, Association for Computing Machinery, New York, NY, USA, 2019.
- [62] F. Zonzini, V. Dertimanis, E. Chatzi, and L. De Marchi, "System identification at the extreme edge for network load reduction in vibration-based monitoring," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20467–20478, 2022.
- [63] T. Cai, P. Liu, D. Niu, J. Shi, and L. Li, "The embedded iot time series database for hybrid solid-state storage system," *Scientific Programming*, vol. 2021, Article ID 9948533, 13 pages, 2021.
- [64] R. Du, S. Magnusson, and C. Fischione, "The internet of things as a deep neural network," *IEEE Communications Magazine*, vol. 58, no. 9, pp. 20–25, 2020.
- [65] R. Vestergaard, Q. Zhang, M. Sipos, and D. E. Lucani, "Titchy: online time-series compression with random access for the internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17568–17583, 2021.
- [66] S. M. Araque, I. Martinez, G. Z. Papadopoulos, N. Montavont, and L. Toutain, "Toward a standard time series representation for IoT based on cbor templates," in *2022 Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 13–19, IEEE, 2022.
- [67] I. R. Panneerselvam, "Transfer learning autoencoder used for compressing multimodal biosignal," *Multimedia Tools and Applications*, vol. 81, pp. 17547–17565, 2022.
- [68] J. Liu, K. Wang, and F. Chen, "TSCache: an efficient flash-based caching scheme for time-series data workloads," *Proceedings of the VLDB Endowment*, vol. 14, no. 13, pp. 3253–3266, 2021.
- [69] A. K. M. Al-Qurabat, S. A. Abdulzahra, and A. K. Idrees, "Two-level energy-efficient data reduction strategies based on SAX-LZW and hierarchical clustering for minimizing the huge data conveyed on the internet of things networks," *The Journal of Supercomputing*, vol. 78, pp. 17844–17890, 2022.
- [70] T. Hafeez and G. McArdle, "Using dynamic perceptually important points for data reduction in IoT," in *Proceedings of the 11th International Conference on the Internet of Things*, pp. 33–39, Association for Computing Machinery, New York, NY, USA, 2021.
- [71] A. L. Michala, I. Vourganas, and A. Coraddu, "Vibration edge computing in maritime IoT," *ACM Transactions on Internet of Things*, vol. 3, no. 1, pp. 1–18, 2021.
- [72] H. A. Dau, E. Keogh, K. Kaveh et al., "The UCR time series classification archive," 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [73] D. Dua and C. Graff, "UCI machine learning repository," 2017, <http://archive.ics.uci.edu/ml>.
- [74] R. Leszczyna, "Aiming at methods' wider adoption: applicability determinants and metrics," *Computer Science Review*, vol. 40, Article ID 100387, 2021.
- [75] M. S. Farooq, S. Riaz, A. Abid, T. Umer, and Y. B. Zikria, "Role of IoT technology in agriculture: a systematic literature review," *Electronics*, vol. 9, no. 2, Article ID 319, 2020.
- [76] H. J. T. Manaligod, M. J. S. Diño, S. Ghose, and J. Han, "Context computing for internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1361–1363, 2020.