



Article

Quality Assurance Technologies of Big Data Applications: A Systematic Literature Review

Shunhui Ji ¹ , Qingqiu Li ¹, Wennan Cao ¹, Pengcheng Zhang ^{1,*}  and Henry Muccini ²

¹ College of Computer and Information, Hohai University, Nanjing 211100, China; shunhuiji@hhu.edu.cn (S.J.); qingqiuli@hhu.edu.cn (Q.L.); cwn@hhu.edu.cn (W.C.)

² Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, 67100 L'Aquila, Italy; henry.muccini@univaq.it

* Correspondence: pchzhang@hhu.edu.cn; Tel.: +86-25-5809-9106

Received: 15 October 2020; Accepted: 9 November 2020; Published: 13 November 2020



Abstract: Big data applications are currently used in many application domains, ranging from statistical applications to prediction systems and smart cities. However, the quality of these applications is far from perfect, such as functional error, failure and low performance. Consequently, assuring the overall quality for big data applications plays an increasingly important role. This paper aims at summarizing and assessing existing quality assurance (QA) technologies addressing quality issues in big data applications. We have conducted a systematic literature review (SLR) by searching major scientific databases, resulting in 83 primary and relevant studies on QA technologies for big data applications. The SLR results reveal the following main findings: (1) the quality attributes that are focused for the quality of big data applications, including correctness, performance, availability, scalability and reliability, and the factors influencing them; (2) the existing implementation-specific QA technologies, including specification, architectural choice and fault tolerance, and the process-specific QA technologies, including analysis, verification, testing, monitoring and fault and failure prediction; (3) existing strengths and limitations of each kind of QA technology; (4) the existing empirical evidence of each QA technology. This study provides a solid foundation for research on QA technologies of big data applications and can help developers of big data applications apply suitable QA technologies.

Keywords: quality attribute; quality assurance technology; big data application; systematic literature review

1. Introduction

The big data technology market grows at a 27% compound annual growth rate (CAGR), and big data market opportunities will reach over 203 billion dollars in 2020 [1]. Big data application systems [2,3], abbreviated as big data applications, refer to the software systems that can collect, process, analyze or predict a large amount of data by means of different platforms, tools and mechanisms. Big data applications are now increasing, being used in many areas, such as recommendation systems, monitoring systems and statistical applications [4,5]. Big data applications are associated with the so-called 4V attributes, e.g., volume, velocity, variety and veracity [6]. Due to the large amount of data generated, the fast velocity of arriving data and the various types of heterogeneous data, the quality of data is far from ideal, which makes the software quality of big data applications far from perfect [7]. For example, due to the volume and velocity attributes [8,9], the data generated from big data applications are extremely numerous and more so with high speed Internet, which may affect data accuracy and data timeliness [10], and consequently lead to software quality problems, such as performance and availability issues [10,11]. Due to the huge variety of heterogeneous

data [12,13], data types and formats are increasingly rich, including structured, semi-structured and unstructured, which may affect data accessibility and data scalability, and hence lead to usability and scalability problems.

In general, quality assurance (QA) is a way to detect or prevent mistakes or defects in manufactured software/products and avoid problems when solutions or services are delivered to customers [14]. However, compared with traditional software systems, big data applications raise new challenges for QA technologies due to the four big data attributes (for example, the velocity of arriving data, and the volume of data) [15]. Many scholars have illustrated current QA problems for big data applications [16,17]. For example, it is a hard task to validate the performance, availability and accuracy of a big data prediction system due to the large-scale data size and the feature of timeliness. Due to the volume and variety attributes, keeping big data recommendation systems scalable is very difficult. Therefore, QA technologies of big data applications are becoming a hot research topic.

Compared with traditional applications, big data applications have the following special characteristics: (a) statistical computation based on large-scale, diverse formats, with structured and non-structured data; (b) machine learning and knowledge-based system evolution; (c) intelligent decision-making with uncertainty; and (d) more complex visualization requirements. These new features of big data applications need novel QA technologies to ensure quality. For example, compared with data in traditional applications (graphics, images, sounds, documents, etc.), there is a substantial amount of unstructured data in big data applications. These data are usually heterogeneous and lack integration. Since the handling of large-scale data is not required in the traditional applications, traditional testing processes lack testing methods for large-scale data, especially in the performance testing. Some novel QA technologies are urgently needed to solve these problems.

In the literature, many scholars have investigated the use of different QA technologies to assure the quality of big data applications [15,18–20]. Some papers have presented overviews on quality problems of big data applications. Zhou et al. [18] presented the first comprehensive study on the quality of the big data platform. For example, they investigated the common symptoms, causes and mitigation strategies of quality issues, including hardware faults, code defects and so on. Juddoo [19] et al. have systematically studied the challenges of data quality in the context of big data. Gao et al. [15] did a profound study on the validation of big data and QA, including the basic concepts, issues and validation process. They also discussed the big data QA focuses, challenges and requirements. Zhang et al. [20] introduced big data attributes and quality attributes; some quality assurance technologies such as testing and monitoring were also discussed. Although these authors have proposed a few QA technologies for big data applications, publications on QA technologies for big data applications remain scattered in the literature, and this hampers the analysis of the advanced technologies and the identification of novel research directions. Therefore, a systematic study of QA technologies for big data applications is still necessary and critical.

In this paper, we provide an exhaustive survey of QA technologies that have significant roles in big data applications, covering 83 papers published from Jan. 2012 to Dec. 2019. The major purpose of this study was to look into literature that is related to QA technologies for big data applications. Then, a comprehensive reference list concerning challenges of QA technologies for big data applications was prepared. In summary, the major contributions of the paper are described in the following:

- The elicitation of big data attributes, and the quality problems they introduce to big data applications;
- The identification of the most frequently used big data QA technologies, together with an analysis of their strengths and limitations;
- A discussion of existing strengths and limitations of each kind of QA technology;
- The proposed QA technologies are generally validated through real cases, which provides a reference for big data practitioners.

Our research resulted in four main findings which are summarized in Table 1.

Table 1. Key findings and implications of this research.

Numbers	Key Findings	Implications
F1	Quality attributes, including correctness, performance, availability, scalability and reliability, are focused for the quality of big data applications and influenced by the big data attributes.	Through our research, we can identify the most important quality attributes that state-of-the-art works address and their influencing factors.
F2	Existing implementation-specific QA technologies include specification, architectural choice and fault tolerance, and process-specific technologies include analysis, verification, testing, monitoring and fault & failure prediction.	Surveying and summarizing existing quality assurance technologies for big data applications.
F3	Existing strengths and limitations of each kind of QA technique.	Through the systematic review, strengths and limitations of each kind of QA technique are discussed and compared.
F4	Existing empirical evidence of each kind of QA technique.	Validating the proposed QA technologies through real cases and providing a reference for big data practitioners.

The findings of this paper contribute general information for future research, as the quality of big data applications will become increasingly more important. They also can help developers of big data applications apply suitable QA technologies. Existing QA technologies have a certain effect on the quality of big data applications; however, some challenges still exist, such as the lack of quantitative models and algorithms.

The rest of the paper is structured as follows. The next section reviews related background and previous studies. Section 3 describes our systematic approach for conducting the review. Section 4 reports the results of themes based on four research questions raised in Section 3. Section 5 provides the main findings of the survey and provides existing research challenges. Section 6 describes some threats in this study. Conclusions and future research directions are given in the final section.

2. Related Work

There is no systematic literature review (including a systematic mapping study, a systematic study and a literature review) that focuses on QA technologies for big data applications. However, quality issues are prevalent in big data [21], and the quality of big data applications has attracted attention and been the focus of research in previous studies. In the following, we first try to describe all the relevant reviews that are truly related to the quality of big data applications.

Zhou et al. [18] presented a comprehensive study on the quality of the distributed big data platform. Massive workloads and scenarios make the data scale keep growing rapidly. Distributed storage systems (GFS, HDFS) and distributed data-parallel execution engines (MapReduce, Hadoop and Dryad) need to improve the processing ability of real-time data. They investigated common symptoms, causes and mitigation measures for quality problems. In addition, there will be different types of problems in big data computing, including hardware failure, code defects, etc. Their discovery is of great significance to the design and maintenance of future distributed big data platform.

Juddoo et al. [19] systematically studied the challenges of data quality in the context of big data. They mainly analyzed and proposed the data quality technologies that would be more suitable for big data in a general context. Their goal was to probe diverse components and activities forming parts of data quality management, metrics, dimensions, data quality rules, data profiling and data cleansing. In addition, the volume, velocity and variety of data may make it impossible to determine the data quality rules. They believed that the measurement of big data attributes is very important to the users' decision-making. Finally, they also listed existing data quality challenges.

Gao and Tao [4,15] first provided detailed discussions for QA problems and big data validation, including the basic concepts and key points. Then they discussed big data applications influenced by big data features. Furthermore, they also discussed big data validation processes, including data collection, data cleaning, data cleansing, data analysis, etc. In addition, they summarized the big data QA issues, challenges and needs.

Zhang et al. [20] further considered QA problems of big data applications, and explored existing methods and tools to improve the quality of distributed platforms and systems; they summarized six QA technologies combined with big data attributes, and explored the big data attributes of existing QA technologies.

Liu et al. [22] pointed out and summarized the issues faced by big data research in data collection, processing and analysis in the current big data area, including uncertain data collection, incomplete information, big data noise, representability, consistency, reliability and so on.

For the distributed systems, Ledmi et al. [23] introduced the basic concept of distributed systems and their main fault types. In order to ensure the stable operation of cluster and grid computing and the cloud, the system needs to have a certain fault tolerance ability. Therefore, various fault-tolerant technologies were discussed in detail, including active fault-tolerance and reactive fault-tolerance. Niedermaier et al. [24] found that the dynamics and complexity of distributed systems bring more challenges to system monitoring. Although there are many new technologies, it is difficult for companies to carry out practical operations. Therefore, they conducted an industry study from different stakeholders involved in monitoring; summarized the existing challenges and requirements; and proposed solutions from various angles.

To sum up, for the big data applications, Zhou et al. [18] and Liu et al. [22] mainly studied the problems and common failures. Gao and Tao [4,15] mainly discussed the QA problems from the perspective of data processing. Juddoo et al. [19] surveyed the topic of data quality technologies. Zhang et al. [20] considered data attributes, QA technologies and existing quality challenges. In addition to the six QA technologies identified in [20], two QA technologies, specification and analysis, are newly added in this paper. Additionally, the QA technologies are introduced and discussed in more detail. Moreover, the strengths and limitations of QA technologies are compared and concluded. We also investigated empirical evidence of QA technologies which can provide references for practitioner.

3. Research Method

In this work, the systematic literature review (SLR) approach proposed by Kitchenham et al. [25] was used to extract QA technologies for big data applications and related questions. Based on the SLR and our research problem, research steps can be raised, as shown in Figure 1. Through these research steps, we can obtain the desired results.

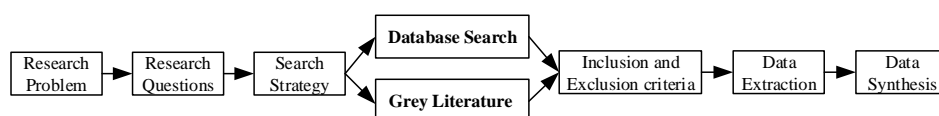


Figure 1. SLR protocol.

3.1. Research Questions

We used the goal-question-metric (GQM) perspectives (i.e., purpose, issue, object and viewpoint) [26] to draw up the aim of this study. The result of the application of the goal-question-metric approach is the specification of a measurement system targeting the given set of problems and a set of rules for understanding the measurement data [27]. Table 2 provides the purpose, issue, object and viewpoint of the research topic.

Table 2. Goal of this research.

Goal	Purpose	Identify, analyze and extract QA technologies
	Issue	for big data applications, and then understand
	Object	the features and challenges of technologies
	viewpoint	in existence from a researcher's viewpoint.

Research questions can usually help us to perform an in-depth study and achieve purposeful research. Based on this research, there are four research questions. Table 3 shows four research questions that we translated from Table 2. RQ1 concentrates on identifying the quality attributes that are focused on in QA of the big data applications and analyzing the factors influencing them. RQ2 concentrates on identifying and classifying existing QA technologies and understanding the effects of them. RQ3 concentrates on analyzing the strengths and limitations of those QA technologies. RQ4 concentrates on validating the proposed QA technologies through real cases and providing a reference for big data practitioners.

Table 3. Research questions.

ID	Research Question
RQ1	Which of the quality attributes are focused in QA of big data applications?
RQ2	Which kinds of technologies are used to guarantee the quality of big data applications?
RQ3	What are the strengths and limitations of the proposed technologies?
RQ4	What are the real cases of using the proposed technologies?

3.2. Search Strategy

The goal of this systematic review is thoroughly examining the literature on QA technologies for big data applications. Three main phases of SLR are presented by EBSE (evidence-based software engineering) [28] guidelines that include planning, execution and reporting results. Moreover, the search strategy is an indispensable part and consists of two different stages.

Stage 1: Database search.

Before we carried out automatic searches, the first step was the definition and validation of the search string to be used for automated search. This process started with pilot searches on seven databases, as shown in Table 4. We combined different keywords that are related to research questions. Table 5 shows the search terms we used in the seven databases, and the search string is defined in the following:

(a AND (b OR c) AND (d OR e OR f OR g)) IN (Title or Abstract or Keyword).

Table 4. Study resources.

Source	Address
ACM Digital Library	http://dl.acm.org/
IEEE Xplore Digital Library	http://ieeexplore.ieee.org/
Springer Link	http://link.springer.com/
Science Direct	http://www.sciencedirect.com/
Scopus	http://www.scopus.com/
Engineering Village	http://www.engineeringvillage.com/
ISI Web of Science	http://isiknowledge.com/

We used a “quasi-gold standard” [29] to validate and guarantee the search string. We use IEEE and ACM libraries as representative search engines to perform automatic searches and refine the search string until all the search items met the requirements and the number of remaining papers was minimal. Then, we used the defined search string to carry out automatic searches. We chose ACM Digital Library, IEEE Xplore Digital Library, Engineering Village, Springer Link, Scopus, ISI Web of Science and Science Direct because those seven databases are the largest and most complete scientific databases that include computer science. We manually downloaded and searched the proceedings if venues were not included in the digital libraries. After the automatic search, a total of 3328 papers were collected.

Table 5. Search terms.

Search ID	Database Search Terms
a	big data
b	application
c	system
d	quality
e	performance
f	quality assurance
g	QA

Stage 2: Gray literature.

To cover gray literature, some alternative sources were investigated as follows:

- Google Scholar

In order to adapt the search terms to Google Scholar and improve the efficiency of the search process, search terms were slightly modified. We searched and collected 1220 papers according to the following search terms:

- (big data AND (application OR system) AND ((quality OR performance OR QA) OR testing OR analysis OR verification OR validation))
- (big data AND (application OR system) AND (quality OR performance OR QA) AND (technique OR method))
- (big data AND (application OR system) AND (quality OR performance OR QA) AND (problem OR issue OR question))
- Checking the personal websites of all the authors of primary studies, in search for other related sources (e.g., unpublished or latest progress).

Through two stages, we found 4548 related papers. Only 102 articles met the selection strategy (discussed below) and are listed in Section 3.3. Then, we scanned all the related results according to the snowball method [30], and we referred to the references cited by the selected paper and included them if they were appropriate. We expanded the number of papers to 121; for example, we used this technique to find [31], which corresponds to our research questions from the references in [32].

To better manage the paper data, we used NoteExpress (<https://noteexpress.apponic.com/>), which is a professional-level document retrieval and management system. Its core functions cover all aspects of "knowledge acquisition, management, application and mining." It is a perfect tool for academic research and knowledge management. However, the number of these results is too large. Consequently, we filtered the results by using the selection strategy described in the next section.

3.3. Selection Strategy

In this subsection, we focus on the selection of research literature. According to the search strategy, much of the returned literature is unnecessary. It is essential to define the selection criteria (inclusion and exclusion criteria) for selecting the related literature. We describe each step of our selection process in the following:

- Combination and duplicate removal. In this step, we sort out the results that we obtain from stage 1 and stage 2 and remove the duplicate content.
- Selection of studies. In this step, the main objective is to filter all the selected literature in light of a set of rigorous inclusion and exclusion criteria. There are five inclusion and four exclusion selection criteria we have defined as described below.
- Exclusion of literature during data extraction. When we read a study carefully, it can be selected or rejected according to the inclusion and exclusion criteria in the end.

When all inclusion criteria are met, the study is selected; otherwise, it is discarded if any exclusion criteria are met. According to the research questions and research purposes, we identified the following inclusion and exclusion criteria.

A study should be chosen if it satisfies the following inclusion criteria:

- (1) The study of the literature focuses on the quality of big data applications or big data systems, in order to be aligned with the theme of our study.
- (2) One or more of our research questions must be directly answered.
- (3) The selected literature must be in English.
- (4) The literature must consist of journal papers or papers published as part of conference or workshop proceedings.
- (5) Studies must have been published in or after 2012. From a simple search (for which we use the search item "big data application") in the EI (engineering index) search library, we can see that most of the papers on big data applications or big data systems were published after 2011, as shown in Figure 2. The literature published before 2012 rarely took into account the quality of big data applications or systems. By reading the relevant literature abstracts, we found that these earlier papers were not relevant to our subject, so we excluded them.

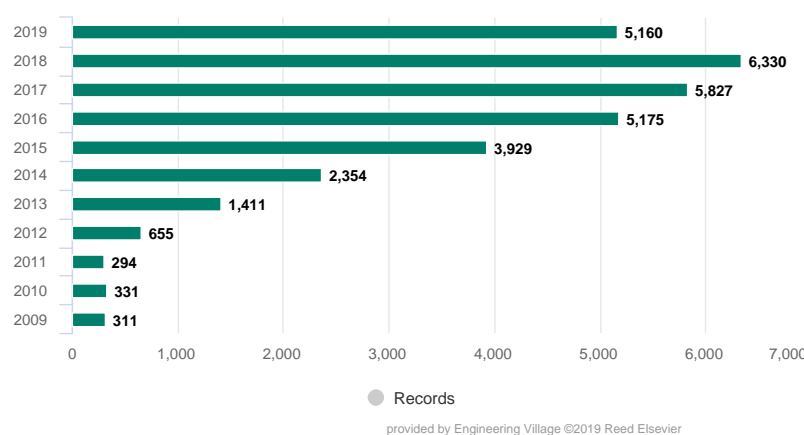


Figure 2. Distribution of papers in EI until Dec 2019.

The main objective of our study was to determine the current technologies of ensuring the quality of big data applications and the challenges associated with the quality of big data applications. This means that the content of the article should be related to the research questions of this paper.

A study should be discarded if it satisfies any one of the following exclusion criteria:

- (1) It is related to big data but not related to the quality of big data applications. Our goal is to study the quality of big data applications or services, rather than the data quality of big data, although data quality can affect application quality.
- (2) It does not explicitly discuss the quality of big data applications and the impact of big data applications or quality factors of big data systems.
- (3) Duplicated literature. Many articles have been included in different databases, and the search results contain repeated articles. For conference papers that meet our selection criteria but are also extended to journal publications, we choose journal publications because they are more comprehensive in content.
- (4) Studies that are not related to the research questions.

Inclusion criteria and exclusion criteria are complementary. Consequently, both the inclusion and exclusion criteria were considered. In this way, we could achieve the desired results. A detailed process of identifying relevant literature is presented in Figure 3. Obviously, the analysis of all the literature presents a certain degree of difficulty. First, by applying these inclusion and exclusion criteria, two researchers separately read the abstracts of all studies selected in the previous step to avoid prejudice as much as possible. Of the initial studies, 488 were selected in this process. Second, for the final selection, we read the entire initial papers and then selected 102 studies. Third, we expanded the number of final studies to 121 according to our snowball method. Conflicts were resolved by extensive discussion. We excluded a number of papers because they were not related and had 83 primary studies at the end of this step.

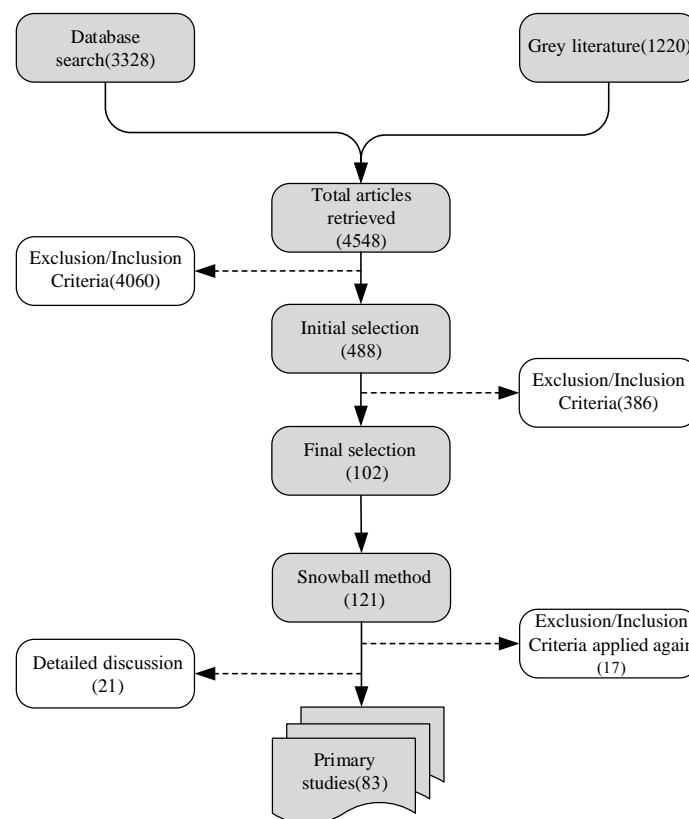


Figure 3. The process of primary study selection.

In the process, the first author and the second author worked together to develop research questions and search strategies, and the second author and four students of the first author executed the search plan together. During the process of finalizing the primary articles, all members of the group had a detailed discussion on whether the articles excluded by only few researchers were in line with our research topic.

3.4. Quality Assessment

After screening the final primary studies by inclusion and exclusion criteria, the criteria for the quality of the study were determined according to the guidelines proposed by Kitchenham and Charters [33]. The corresponding quality checklist is shown in Table 6. The table includes 12 questions that consider four research quality dimensions, including research design, behavior, analysis and conclusions. For each quality item we set a value of 1 if the authors put forward an explicit description, 0.5 if there was a vague description and 0 if there was no description at all. The author and his research assistant applied the quality assessment method to each major article, compared the results and discussed any differences until a consensus was reached. We scored each possible answer for each question in the main article and converted it into a percentage after coming to an agreement. The presentation quality assessment results of the preliminary study indicate that most studies have deep descriptions of the problems and their background, and most studies have fully and clearly described the contributions and insights. Nevertheless, some studies do not describe the specific division of labor in the method introduction, and there is a lack of discussion of the limitations of the proposed method. However, the total average score of 8.8 out of 12 indicates that the quality of the research is good, supporting the validity of the extracted data and the conclusions drawn therefrom.

Table 6. Quality assessment questions and results.

ID	Question	Percentage		
		Yes	Partially	No
Design				
Q1	Are the aims of the study clearly stated?	100%	0%	0%
Q2	Are the chosen quality attributes distinctly stated and defined?	55.4%	43.2%	0.4%
Q3	Was the sample size reasonable?	32.4%	45.9%	21.7%
Conduct				
Q4	Are research methods adequately described?	90.5%	9.5%	0%
Q5	Are the datasets completely described (source, size, and programming languages)?	32.4%	43.2%	24.4%
Q6	Are the observation units or research participants described in the study?	2.7%	0%	97.3%
Analysis				
Q7	Is the purpose of the analysis clear?	98.6%	1.4%	0%
Q8	Are the statistical methods described?	14.9%	5.4%	79.7%
Q9	Is the statistical significance of the results reported?	14.9%	5.4%	79.7%
Conclusion				
Q10	Are the results compared with other methods?	27.0%	1.4%	71.6%
Q11	Do the results support the conclusions?	100%	0%	0%
Q12	Are validity threats discussed?	18.9%	28.4%	52.7%

3.5. Data Extraction

The goal of this step was to design forms to identify and collect useful and relevant information from the selected primary studies so that it could answer our research questions proposed in Section 3.1. To carry out an in-depth analysis, we could apply the data extraction form to all selected primary studies. Table 7 shows the data extraction form. According to the data extraction form, we collected specific information in an excel file (<https://github.com/QXL4515/QA-techniques-for-big-data-application>). In this process, the first author and the second author jointly developed an information extraction strategy to lay the foundation for subsequent analysis. In addition, the third author validated and confirmed this research strategy.

3.6. Data Synthesis

Data synthesis is used to collect and summarize the data extracted from primary studies. Moreover, the main goal is to understand, analyze and extract current QA technologies for big data applications. Our data synthesis was specifically divided into two main phases.

Phase 1: We analyzed the extracted data (most of which are included in Table 7 and some were indispensable in the research process) to determine the trends and collect information about our research questions and record them. In addition, we classified and analyzed articles according to the research questions proposed in Section 3.1.

Phase 2: We classified the literature according to different research questions. The most important task was to classify the articles according to the QA technologies through the relevant analysis.

Table 7. Data extraction form.

Data Item	Extracted Data	Description	Type
1	Study title	Reflect the relevant research direction	Whole research
2	Publication year	Indicating the trend of research	Whole research
3	Journal/Conference	The type of study: the conference or the journal	Whole research
4	Authors	Research author's other relevant studies	Whole research
5	Context study	Understand the full text	Whole research
6	Existing challenges	The limitations of the approaches and the challenges of big data application	Whole research
7	Big data attributes	Related 4V attributes	RQ1
8	Quality requirements	The attributes of the demand	RQ1
9	Quality attributes	Quality attributes of big data application	RQ1
10	Technology	Application technology	RQ2
11	Quality assurance technologies	QA technology	RQ2
12	Experimental results	The effectiveness of the methods	RQ2
13	Strengths	The advantages of the approaches	RQ3
14	Empirical Evidence	Real cases of the methods	RQ4

4. Results

This section, deeply analyzing the primary studies provides an answer to the four research questions presented in Section 3.

In addition, Figures 4–6 provide some simple statistics. Figure 4 presents how our primary studies are distributed over the years. Figure 5 groups the primary studies according to the type of publication. Figure 6 counts the number of studies retrieved from different databases.

While Section 4.1 provides an overview of the main concepts discussed in this section, Sections 4.2–4.5 report the answer to the research questions.

4.1. Overview of the Main Concepts

While answering the four research questions identified in previous sections, we will utilize and correlate three different dimensions: big data attributes, data quality parameters and software quality attributes. (The output of this analysis is reported in Section 5.2.)

Big data attributes: Big data applications are associated with the so-called 4V attributes, e.g., volume, velocity, variety and veracity [6]. In this study, we take into account only three of the 4V big data attributes (excluding the veracity one) for the following reasons: First, through the initial reading of the literature, many papers are not concerned about the veracity. Second, big data currently have multi-V attributes; only three attributes (volume, variety and velocity) are recognized extensively [34,35].

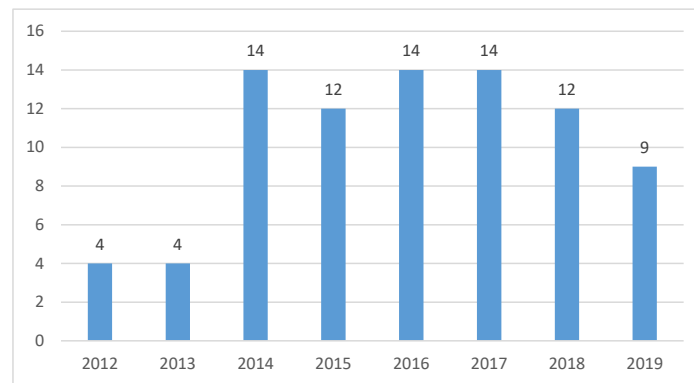


Figure 4. Distribution of papers during 2012–2019.

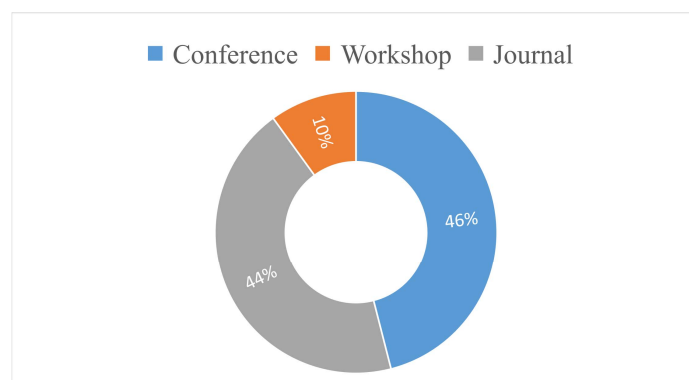


Figure 5. Distribution of papers by literature types.

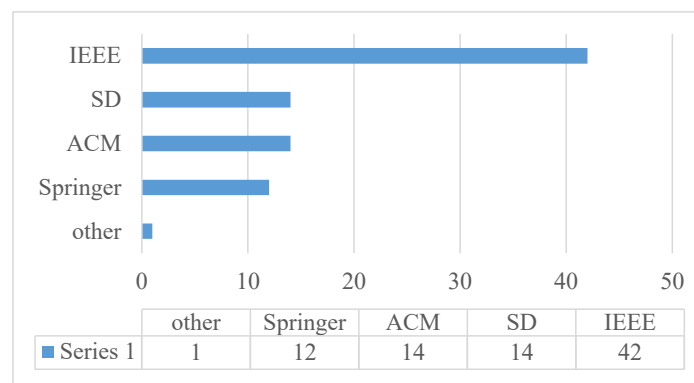


Figure 6. Distribution of papers by databases.

Data quality parameters: Data quality parameters describe the measure of the quality of data. Since data are an increasingly vital part of applications, data quality becomes an important concern. Poor data quality could affect enterprise revenue, waste company resources, introduce lost productivity and even lead to wrong business decisions [15]. According to the Experian Data Quality global benchmark report (Erin Haselkorn, "New Experian Data Quality research shows inaccurate data preventing desired customer insight", Posted on Jan 29 2015 at URL <http://www.experian.com/blogs/news/2015/01/29/data-quality-research-study/>), U.S. organizations claim that 32 percent of their data are wrong on average. Since data quality parameters are not universally agreed upon, we extracted them by analyzing papers [15,36,37].

Software quality attributes: Software quality attributes describe the attributes that software systems shall expose. We started from the list provided in the ISO/IEC 25010:2011 standard and

selected those quality attributes that are mostly recurrent in the primary studies. A quality model consists of five characteristics (correctness, performance, availability, scalability and reliability) that relate to the outcome of the interaction when a product is used in a particular context. This system model is applicable to the complete human–computer system, including both computer systems in use and software products in use.

QA technologies: Eight technologies (specification, analysis, architectural choice, fault tolerance, verification, testing, monitoring, fault and failure prediction) have been developed for the static attributes of software and dynamic attributes of the computer system. They can be classified into two types: implementation-specific technologies and process-specific technologies.

The implementation-specific QA technologies include:

- Specification: A specification refers to a type of technical standard. The specification includes requirement specification, functional specification and non-functional specification of big data applications.
- Architectural choice: Literature indicates that QA is achieved by selecting model-driven architecture (MDA) for helping developers creating software that meets non-functional properties, such as, reliability, safety and efficiency. As practitioners chose MDA and nothing else, it can be concluded that this solution is superior to other architectural choices.
- Fault tolerance: Fault tolerance technique is used to improve the capability of maintaining performance of the big data application in cases of faults, such as correctness, reliability and availability.

The process-specific QA technologies include:

- Analysis: The analysis technique is used to analyze the main factors which can affect the big data applications' quality, such as performance, correctness and others, which plays an important role in the design phase of big data applications.
- Verification: Verification relies on existing technologies to validate whether the big data applications satisfy desired quality attributes. Many challenges for big data applications appear due to the 3V properties. For example, the volume and the velocity of data may make it a difficult task to validate the correctness attribute of big data applications.
- Testing: Different kinds of testing techniques are used to validate whether big data applications conform to requirement specifications. Using this technique, inconsistent or contradictory with the requirement specifications can be identified.
- Monitoring: Runtime monitoring is an effective way to ensure the overall quality of big data applications. However, runtime monitoring may occur additional loading problems for big data applications. Hence, it is necessary to improve the performance of big data monitoring.
- Fault and failure prediction: Big data applications faces many failures. If the upcoming failure can be predicted, the overall quality of big data applications may be greatly improved.

4.2. Identify the Focused Quality Attributes in Big Data Applications (RQ1)

The goal of this section is to answer RQ1 (which of the quality attributes are focused in QA of big data applications?). Besides identifying the quality attributes focused in the primary studies, we also analyze the influencing factors brought by the big data attributes, so that the factors can be taken into consideration in QA of big data applications.

We use the ISO/IEC 25010:2011 to extract the quality attributes of big data applications. Table 8 provides the statistical distribution of different quality attributes. According to statistics, we identify related quality attributes, as shown in Figure 7. For some articles that may involve more than one quality attribute, such as [38,39], we choose the main quality attribute that they convey.

Table 8. Distribution of the quality attributes.

Attributes	Studies
Correctness	[31,40–53]
Performance	[17,38,54–70]
Availability	[63,71–88]
Scalability	[89–93]
Reliability	[32,39,53,62,94–105]
Others	[106–113]

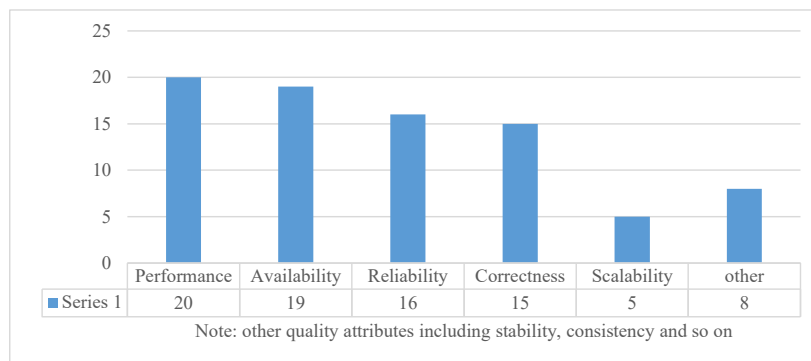


Figure 7. The quantity distribution of the quality attributes.

From the 83 primary studies, some quality attributes are discussed, including correctness, performance, availability, scalability, reliability, efficiency, flexibility, robustness, stability, interoperability and consistency.

From Figure 7, we can see that the five main quality attributes have been discussed in the 83 primary studies. However, there are fewer articles focused on other attributes, such as stability, consistency, efficiency and so on.

- **Correctness:** Correctness measures the probability that big data applications can “get things right.” If the big data application cannot guarantee the correctness, then it will have no value at all. For example, a weather forecast system that always provides the wrong weather is obviously not of any use. Therefore, correctness is the first attribute to be considered in big data applications. The papers [41,44] provide the fault tolerance mechanism to guarantee the normal operation of applications. If the big data application runs incorrectly, it will cause inconvenience or even loss to the user. The papers [40,45] provide the testing method to check the fault of big data applications to assure the correctness.
- **Performance:** Performance refers to the ability of big data applications to provide timely services, specifically in three areas, including the average response time, the number of transactions per unit time and the ability to maintain high-speed processing. Due to the large amounts of data, performance is a key topic in big data applications. In Table 8, we show the many papers that refer to the performances of big data applications. The major purpose of focusing on the performance problem is to handle big data with limited resources in big data applications. To be precise, the processing performance of big data applications under massive data scenarios is its major selling point and breakthrough. According to the relevant literature, we can see that common performance optimization technologies for big data applications are generally divided into two parts [44,55,56,71,72,94]. The first one consists of hardware and system-level observations to find specific bottlenecks and make hardware or system-level adjustments. The second one is to achieve optimization mainly through adjustments to specific software usage methods.

- **Availability:** Availability refers to the ability of big data applications to perform a required function under stated conditions. The rapid growth of data has made it necessary for big data applications to manage data streams and handle an impressive volume, and since these data types are complex (variety), the operation process may create different kinds of problems. Consequently, it is important to ensure the availability of big data applications.
- **Scalability:** Scalability refers to the ability of large data applications to maintain service quality when users and data volumes increase. For a continuous stream of big data, processing systems, storage systems, etc., should be able to handle these data in a scalable manner. Moreover, the system would be very complex for big data applications. For better improvement, the system must be scalable. Paper [89] proposes a flexible data analytic framework for big data applications, and the framework can flexibly handle big data with scalability.
- **Reliability:** Reliability refers to the ability of big data applications to apply the specified functions within the specified conditions and within the specified time. Reliability issues are usually caused by unexpected exceptions in the design and undetected code defects. For example, paper [39] uses a monitoring technique to monitor the operational status of big data applications in real time so that failures can occur in real time and developers can effectively resolve these problems.

Although big data applications have many other related quality attributes, the focal ones are the five mentioned above. Therefore, these five quality attributes are critical to ensuring the quality of big data applications and are the main focus of this survey.

To better perform QA of the focal quality attributes, the factors that may affect the quality attributes should be identified. We extracted the relationships existing among the big data attributes, the data quality parameters and the software quality attributes to analyze the influencing factors brought by big data attributes. We have finalized five data quality parameters, including data correctness, data completeness, data timeliness, data accuracy and data consistency. Data correctness refers to the correctness of data in transmission, processing and storage. Data completeness is a quantitative measurement that is used to evaluate how much valid analytical data is obtained compared to the planned number [15] and is usually expressed as a percentage of usable analytical data. Data timeliness refers to real-time and effective data processing [106]. Data accuracy refers to the degree to which data is equivalent to the corresponding real value [107]. Data consistency is useful to evaluate the consistency of given datasets from different perspectives [43]. The relationships that we found on the primary studies are reported in Table 9 and discussed below.

- **Volume:** The datasets used in industrial environments are huge, usually measured in terabytes, or even exabytes. According to [17,114], the larger the volume of the data, the greater the probability that the data will be modified, deleted and so on. In general, increased data reduces data completeness [41,107]. How to deal with a massive amount of data in a very short time is a vast challenge. If these data cannot be processed in a timely manner, the value of these data will decrease and the original goal of building big data systems will be lost [38]. With incorrect data, correctness and availability can not be assured. Application performance will decline as data volume grows. When the amount of data reaches a certain size, the application crashes and cannot provide mission services [38], which also affects reliability. Additionally, the volume of big data attributes will inevitably bring about the scalability issue of big data applications.
- **Velocity:** With the flood of data generated quickly from smart phones and sensors, rapid analysis and processing of data need to be considered [71]. These data must be analyzed in time because the velocity of data generation is very quick [41,115]. Low-speed data processing may result in the fact that the big data systems are unable to respond effectively to any negative changes (speed) [94]. Therefore, the velocity engenders challenges to data timeliness and performance. Data in the high-speed transmission process will greatly increase the data failure rate. Abnormal or missing data will affect the correctness, availability and reliability of big data applications [107].
- **Variety:** The increasing number of sensors that are deployed on the Internet makes the data generated complex. It is impossible for human beings to write every rule for each type

of data to identify relevant information. As a result, most of the events in these data are unknown, abnormal and indescribable. For big data applications, the sources of data are varied, including structured, semi-structured and unstructured data. Some of these data have no statistical significance, which greatly influences data accuracy [54]. Unstructured data will produce the consistency problem [17]. In addition, the contents of the database became corrupted by erroneous programs storing incorrect values and deleting essential records. It is hard to recognize such quality erosion in large databases, but over time, it spreads similarly to a cancerous infection, causing ever-increasing big data system failures. Thus, not only data quality but also the quality of applications suffer under erosion [114].

Table 9. Relations among big data attributes, data quality parameters and software quality attributes.

Big Data Attribute	Data Quality Parameter	Software Quality Attribute
Volume	Data Correctness, completeness, Timeliness	Data Correctness, Availability, Reliability, Performance, Scalability
Velocity	Data Timeliness, Correctness	Data Correctness, Availability, Reliability, Performance
Variety	Data Accuracy, Consistency	Data Correctness, Availability, Reliability, Performance

4.3. Technologies for Assuring the QA of Big Data Applications (RQ2)

This section answers RQ2 (which kinds of technologies are used to guarantee the quality of big data applications?). We extracted the quality assurance technologies used in the primary articles. In Figure 8, we show the distribution of papers for these different types of QA technologies. These technologies cover the entire development process for big data applications. According to the papers we collected, we identified the existing three implementation-specific technologies, including specification; architectural choice and fault tolerance; and five process-specific technologies, including analysis, verification, testing, monitoring and fault and failure prediction.

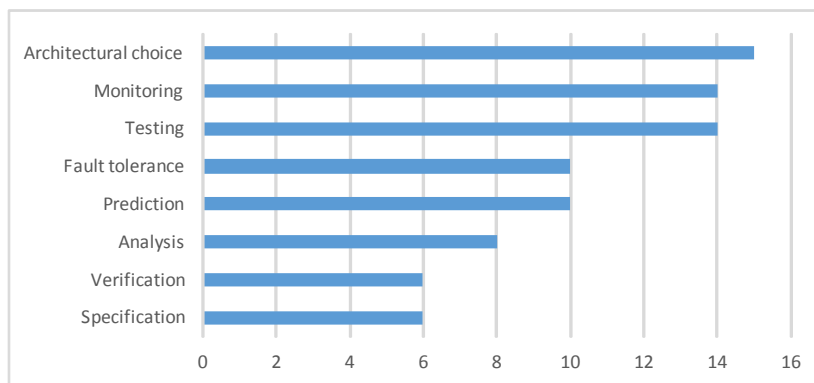


Figure 8. Distribution of Different QA Technologies.

In fact, quality assurance is an activity that applies to the entire big data application process. The development of big data applications is a systems engineering task that includes requirements, analysis, design, implementation and testing. Accordingly, we mainly divide the QA technologies into design time and runtime. Based on the papers we surveyed and the development process of big data applications, we divide the quality assurance technologies for big data applications into the above eight types. In Table 10, we have listed the simple descriptions of different types of QA technologies for big data applications.

We compare the quality assurance technologies from five different aspects, including suitable stage, application domain, quality attribute, effectiveness and efficiency.

Suitable stage refers to the possible stage using these quality assurance technologies, including design time, runtime or both. Application domain means the specific area that big data applications belong to, including DA (database application), DSS (distributed storage system), AIS (artificial intelligence system), BDCS (big data classification system), GSCS (geographic security control system), BS (bioinformatics software), AITS (automated IT system) and so on. Quality attribute identifies the most used quality attributes being addressed by the corresponding technologies. Effectiveness means that big data application quality assurance technologies can guarantee the quality of big data applications to a certain extent. Efficiency refers to the improvement of the quality of big data application through the QA technologies. For better comparisons, we present the results in Table 11 (the QA are those presented in Table 8).

Table 10. Identified QA technologies.

Type	QA Technologies	Description	References
implementation-specific	Specification	A type of technical standard to ensure the quality in the design time.	[61,63,75,90,96,110]
	Architectural Choice	Use MDA to standardize a platform-independent application and select a specific implementation platform for the application.	[69,70,72,74,78,79,86,92,95,97,98,111–113]
	Fault Tolerance	An effective means to address reliability and availability concerns of big data applications.	[41,42,44,46,55,73,80,99,100,106]
process-specific	Analysis	Analyze the quality attributes of big data applications to determine the main factors that affect their quality.	[32,60,65,68,76,77,89,109]
	Verification	Verify whether the big data applications satisfy desired quality attributes.	[43,58,59,62,81,101]
	Testing	Acknowledge application levels of correctness, performance and other quality attributes, and check the testability of big data applications.	[17,38,40,45,47–49,82–84,87,102,103,114]
	Monitoring	Detect failures or potential anomalies at runtime.	[39,50,54,56,64,66,71,88,91,93,94,104,107,116]
	Fault and Failure Prediction	Achieve the prediction of the performance status and potential anomalies of big data applications and provide the information for real-time control.	[31,51–53,57,67,85,105,108,115]

Table 11. Comparison of QA technologies.

Technologies	Stage	Application Domain	Quality Attributes	Effectiveness	Efficiency
Specification	Design time	General	Efficiency, Performance, Scalability	Ensure all requirements be met.	Normally achieve high efficiency.
Analysis	Design time	General	Performance, Scalability, Flexibility	Analyze the quality impact factors as much as possible	Varying degrees can be achieved.
Architectural Choice	Design time	Hadoop, MapReduce, DA	Reliability, Efficiency, Performance, Availability	The model determines the quality of the follow-up.	Achieve high efficiency.
Fault Tolerance	Design and Runtime	Hadoop, DSS, AIS	Performance, Correctness, Reliability, Scalability	Allow to tolerate the occurrence of mistakes within a certain range.	Bring implication on efficiency.
Verification	Design and Runtime	BDCS, MapReduce	Correctness, Reliability, Performance	Identify features that should not be implemented.	Depend on the method.
Testing	Design and Runtime	GSCSystem, BS	Correctness, Scalability, Performance, Scalability	Detect errors, measure software quality, and evaluate whether design requirements are met.	Depend on the method.
Monitoring	Runtime	Hadoop, MapReduce, General	Performance, Availability, Reliability	Effectively monitor the occurrence of errors.	Cause high load.
Fault and Failure Prediction	Design and Runtime	Cloud platforms, AITS	Reliability, Performance	Predict errors that may occur in advance.	Varying degrees can be achieved.

4.4. Existing Strengths and Limitations (RQ3)

The main purpose of RQ3 (what are the advantages and limitations of the proposed technologies?) is to comprehend the strengths and limitations of QA approaches. To answer RQ3, we discuss the strengths and limitations of each technique.

From Table 11, we can see that specification, analysis and architectural choice are used at design time; monitoring is used at runtime; fault tolerance, verification, testing and fault and failure prediction cover both design time and runtime. Design-time technologies are commonly used in MapReduce, which is a great help when designing big data application frameworks. The runtime technologies are usually used after the generation of big data applications, and their applications are very extensive, including intelligent systems, storage systems, cloud computing, etc.

For quality attributes, while most technologies can contend with performance and reliability, some technologies focus on correctness, scalability, etc. To a certain extent, these eight quality assurance technologies assure the quality of big data applications during their application phase, although their effectiveness and efficiencies are different. To better illustrate these three quality parameters, we carry out a specific analysis through two quality assurance technologies. Specification establishes complete descriptions of information, detailed functional and behavioral descriptions and performance requirements for big data applications to ensure the quality. Therefore, it can guarantee the integrity of the function of big data applications to achieve the designated goal of big data applications and guarantee satisfaction. Although it does not guarantee the quality of the application in runtime, it guarantees the quality of the application in the initial stage. As a vital technology in the QA of big data applications, the main function of testing is to test big data applications at runtime. As a

well-known concept, the purpose of testing is to detect errors and measure quality, thereby ensuring effectiveness. In addition, the efficiency of testing is largely concerned with testing methods and testing tools. The detailed description of other technologies is shown in Table 11.

The strengths and limitations of QA technologies are as follows:

- **Specification:** Due to the large amounts of data generated for big data applications, suitable specifications can be used to select the most useful data at hand. This technology can effectively improve the efficiency, performance and scalability of big data applications by using UML [96], ADL [63] and so on. The quality of the system can be guaranteed at the design stage. In addition, the specification is also used to ensure that system functions in big data applications can be implemented correctly. However, all articles are aimed at specific applications or scenarios, and do not generalize to different types of big data applications [63,90].
- **Architectural choice:** MDA is used to generate and export most codes for big data applications and can greatly reduce human error. To date, MDA metamodels and model mappings are only targeted for very special kinds of systems, e.g., MapReduce [72,74]. Metamodels and model mapping approaches for other kinds of big data applications are also urgently needed.
- **Fault tolerance:** Fault tolerance is one of the staple metrics of quality of service in big data applications. Fault-tolerant mechanisms permit big data applications to allow or tolerate the occurrence of mistakes within a certain range. If a minor error occurs, the big data application can still offer stable operation [99,106]. Nevertheless, fault tolerance cannot always be optimal. Furthermore, fault tolerance can introduce performance issues, and most current approaches neglect this problem.
- **Analysis:** The main factors that affect the big data applications' quality analysis are the size of the data, the speed of data processing and data diversity. Analysis technologies can analyze major factors that may affect the quality of software operations during the design phase of big data applications. Current approaches only focus on analyzing performance attributes [60,89]. There is a need to develop approaches for analyzing other quality attributes. In addition, it is impossible to analyze all quality impact factors for big data applications. The specific conditions should be specified before analysis.
- **Verification:** Due to the complexity of big data applications, there is no uniform verification technology in general. Verification technologies verify or validate quality attributes by using logical analysis, a theorem proving and model checking. There is a lack of formal models and corresponding algorithms to verify attributes of big data applications [58,59]. Due to the existence of big data attributes, traditional software verification standards no longer meet the quality requirements [117].
- **Testing:** In contrast to verification, the testing technique is always performed during the execution of big data applications. Due to the large amounts of data, automatic testing is an efficient approach for big data applications. Current research always applies traditional testing approaches in big data applications [47]. However, novel approaches for testing big data attributes are urgently needed because testing focuses are different between big data application testing and conventional testing. Conventional testing focuses on diverse software errors regarding structures, functions, UI and connections to the external systems. In contrast, big data application testing focuses on involute algorithms, large-scale data input, complicated models and so on. Furthermore, conventional testing and big data application testing are different in the test input, the testing execution and the results. As an example, learning-based testing approaches [52] can test the velocity attributes of big data applications.
- **Monitoring:** Monitoring can obtain accurate status and behavioral information for big data applications in a real operating environment. For big data applications running in complex and variable network environments, their operating environments will affect the operations of the software system and produce some unexpected problems. Therefore, monitoring technologies will be more conducive to the timely response to the emergence of anomalies to prevent failures [71,91].

A stable and reliable big data application relies on monitoring technologies that not only monitor whether the service is alive or not but also monitor the operation of the system and data quality. The high velocity of big data engendered the challenge of monitoring accuracy issues and may produce overhead problems for big data applications.

- Fault and failure prediction: Prediction technologies can predict errors that may occur in the operation of big data applications so that errors can be prevented in advance. Due to the complexity of big data applications, the accuracy of prediction is still a substantial problem that we need to consider in the future. Deep learning-based approaches [44,46,55] can be combined with other technologies to improve prediction accuracy due to the large amounts of data.

4.5. Empirical Evidence (RQ4)

The goal of RQ4 (what are the real cases of using the proposed technologies?) is to elicit empirical evidence on the use of QA technologies. We organize the discussion of the QA technologies discussed in Section 4.3, which is shown in Table 12.

- Specification. The approach in [63] is explained by a case study of specifying and modeling a vehicular ad hoc network (VANET). The major merits of the posed method are its capacity to take into consideration big data attributes and cyber physical system attributes through customized concepts and models in a strict, simple and expressive approach.
- Architectural choice. In [74], the investigators demonstrate the effectiveness of the proposed approach by using a case study. This approach can overcome accidental complexities in analytics-intensive big data applications. Paper [97] conducted a series of tests using Amazon's AWS cloud platform to evaluate the performance and scalability of the observable architecture by considering the CPU, memory and network utilization levels. Paper [65] uses a simple case study to evaluate the proposed architecture and a metamodel in the word count application.
- Fault tolerance. The experiments in paper [55] show that DAP architecture can improve the performance of joining two-way streams by analyzing the time consumption and recovery ratio. In addition, all data can be reinstated if the newly started VMs can be reinstated in a few seconds while nonadjacent nodes fail; meanwhile, if neighboring nodes fail, some data can be reinstated. Through analyzing the CPU utilization, memory footprint, disk throughput and network throughput, experiments in paper [46] show that the performance of all cases (MapReduce data computing applications) can be significantly improved.
- Analysis. The experiments in [89] show that the two factors that are the most important concern the quality of scientific data compression and remote visualization, which are analyzed by latency and throughput. Experiments in [60] were conducted to analyze the connection between the performance measures of several MapReduce applications and performance concepts, such as CPU processing time. The consequences of performance analysis illustrate that the major performance measures are processing time, job turnaround and so on. Therefore, in order to improve the performances of big data applications, we must take into consideration these measures.
- Verification. In [43], the author used CMA (cell morphology assay) as an example to describe the design of the framework. Verifying and validating datasets, software systems and algorithms in CMA demonstrates the effectiveness of the framework.
- Testing. In [38], the authors used a number of virtual users to simulate real users and observed the average response time and CPU performance in a network public opinion monitoring system. In [49], the experiment verifies the effectiveness and correctness of the proposed technique in alleviating the Oracle problem in a region growth program. The testing method successfully detects all the embedded mutants.
- Monitoring. The experiments in [94] show that a large queue can increase the write speed and that the proposed framework supports a very high throughput in a reasonable amount of time in a cloud monitoring system. The authors also provide comparative tests to show the effectiveness

- of the framework. In [54], the comparison experiment shows that the method is reliable and fast, especially with the increase of the data volume, and the speed advantage is obvious.
- **Fault and failure prediction.** In [108], the authors implemented the proactive failure management system and tested the performance in a production cloud computing environment. Experimental results show that the approach can reach a high true positive rate and a low false positive rate for failure prediction. In [115], the authors provide emulation-based evaluations for different sets of data traces, and the results show that the new prediction system is accurate and efficient.

Table 12. Experimental summary and statistics.

Techniques	Ref	Case Type	Domain	Metric	Experimental Results
Specification	[63]	Small	Traffic forecasting system	Delay time	Rigorous, easy and expressive
	[74]	Small	Analytics-intensive big data applications	Accidental complexities, Cycle	The effectiveness of MDD for accidental complexities
Architectural choice	[97]	Small	Not mentioned	CPU, memory, network utilization levels	Improve the scalability
	[65]	Small	Word Count application	Not mentioned	High degree of automation
Fault tolerance	[55]	Real-world	Join bidirectional data streams	Time Consuming, Recover Ratio	Improve the performance of joining two-way streams
	[46]	Real-world	MapReduce data computing applications	CPU utilization, Memory footprint, Disk throughput, Network throughput	Transparently enable fault tolerance for applications
Analysis	[89]	Real-world	Scientific data compression and remote visualization	Latency, Throughput	Obtain two factors
	[60]	Large	MapReduce application	Processing time, Job turnaround, Hard disk bytes written	Improve the performance
Verification	[43]	Small	Cell Morphology Assay	MRs	Its effectiveness for testing ADDA
Testing	[38]	Small	Network public opinion monitoring system	Response time	No specific instructions
	[49]	Small	Image Processing	Error detection rate	Detects all the embedded mutants
Monitoring	[94]	Real-world	Cloud monitoring system	Insert operation time	Achieves a response time of a few hundred
	[54]	Real-world	Big data public opinion monitoring platform	Accuracy Rate, Elapsed time	High accuracy and meeting the requirements of real time
	[116]	Large	No specific instructions	Throughput, Read latency, Write latency	Improve the performance by changing various tuning parameters
	[39]	Small	Big data-based condition monitoring of power apparatuses	No specific instructions	Improve the accuracy of condition monitoring
Prediction	[108]	Small	Cloud computing system	False Positive Rate, true positive rate	Achieve high true positive rate, low false positive rate for failure prediction
	[115]	Small	Different applications	Prediction accuracy	New prediction system is accurate and efficient

5. Discussion

The key findings are already provided in Table 1. In this Section, we mainly discuss the cross-cutting findings and existing challenges of this review.

5.1. Cross-Cutting Findings

This subsection discusses some cross-cutting findings deduced from the key findings.

- Relations between big data attributes and quality attributes. The collected results based on main findings show that big data attributes sometimes have contradictory impacts on quality attributes. Some big data attributes are found to improve some quality attributes and weaken others. These findings lead to a conclusion that big data attributes do not always improve all quality attributes of big data applications. To a certain extent, this conclusion matches the definition of big data attributes stated by most of the researchers involved in a study regarding the challenges and benefits between big data attributes and quality attributes in practice.
- Relations among big data attributes, quality attributes and big data applications. In this study, researchers have proposed some quality attributes to effectively assess the impacts of big data attributes on applications. Therefore, we believe that it is incorrect to limit the research on the quality of big data applications to a certain big data property, obtain some negative results and then state a general conclusion that comprehensive consideration of big data attributes causes big data applications' quality to weaken. For example, considering that the data that the system need to process has large volume, high velocity and huge variety, a number of companies have built sophisticated monitoring and analyzing tools that go far beyond simple resource utilization reports. The monitoring systems may occupy too many system resources and degrade the monitored big data application system's performance [116]. Consequently, most big data applications that take into account big data attributes can cause the system to be more complex. We believe that it is incorrect to draw a general conclusion that comprehensive consideration of big data attributes negatively affects big data applications' quality. Moreover, such a conclusion does not consider other quality attributes, such as reliability, scalability and correctness.
- Relations between quality attributes and QA technologies. It is important to note that researchers may also use different QA technologies when considering the same quality attributes. That is, there may be some empirical experience enlisted in practice. It can be inferred that the relation between quality attributes and QA technologies is not one-to-one. For example, correctness can be achieved through a variety of QA technologies, including fault tolerance, verification, testing, monitoring and fault and failure prediction, as analyzed from Tables 8 and 10. On the other hand, when using the same QA technology, different researchers design different methods and evaluation indicators. Therefore, when a study finds that there is a negative or a positive relation between quality attributes and QA technologies, we cannot conclude a specific finding regarding the relation between them. We need to consider investigating this problem for various types of big data applications. For example, for big data applications concerned with processing time, monitoring technologies can improve the performance of the system, but in some common big data applications, an excessive emphasis on monitoring technologies may degrade the performance of the system. Therefore, the specifications of QA technologies and the relationships between QA technologies and quality attributes need to be further studied.

5.2. Existing Challenges

Based on the key findings and cross-cutting findings aforementioned, we discuss some research challenges in this subsection.

- Challenge 1: The problems brought by big data attributes.

Despite that many technologies have been proposed to address big data attributes, existing technologies cannot provide adequate scalability and face major difficulties. Based on the SLR results, Table 13 summarizes the challenges and the possible solutions for 3V attributes. For example, the distributed file system has a high fault tolerance, high throughput and other excellent characteristics. It can use multiple storage servers to share the storage load to store a large amount of data and support linear expansion. When the storage space is insufficient, it can use hot swap to increase storage devices and expand conveniently. These capabilities address the storage and scalability challenges of big data applications caused by the volume attribute. Many studies [118,119] show that distributed file systems can handle large-scale data very well.

For large-scale optimization and high-speed data transmission of big data applications, a decomposition-based distributed parallel programming algorithm [120] is proposed and an online algorithm is designed to dynamically adjust data partitioning and aggregation. Dobre et al. [121] reviewed various parallel and distributed programming paradigms, analyzed how they fit into the big data era and presented modern emerging paradigms and frameworks. Consequently, parallel programming is particularly effective in big data applications, especially for addressing the velocity of data. In addition, NoSQL [122] databases are created to solve the challenges brought by the multiple data types of large-scale data collection, especially the big data application problems. NoSQL's flexible storage structure fundamentally solves the problem of variety and unstructured data storage [41]. At the same time, distributed file systems solve the problem of data storage and greatly reduce costs. It can be seen that these technologies can be combined with existing QA technologies for big data applications in the future.

Table 13. Properties, challenges and technologies.

Properties	Challenge	Possible Solutions
Volume	Storage/Scale	Distributed File Systems
Velocity	Fast Processing	Parallel Programming
Variety	Heterogeneity	NoSQL Databases

- Challenge 2: Lack of the awareness and good understanding of QA technologies for big data applications.

As mentioned in Section 5.1, because different professional skills and understandings of the field exist, big data practitioners tend to use different QA technologies when considering the same quality attributes; therefore, the QA technologies that are chosen according to experience may not be the most appropriate. Moreover, an incorrect application of QA technologies may cause extensive losses. For example, because of an incorrect transaction algorithm, the electronic trading system led to the purchase of 150 different stocks at a low price by the United States KCP (Knight Capital Group) financial companies, resulting in the company suffering a loss of 440 million US dollars with the day shares falling 62% (<https://dealbook.nytimes.com/2012/08/02/knight-capital-says-trading-mishap-cost-it-440-million/>). Therefore, a clear understanding of QA technologies can reduce the implementation of incorrect algorithms and technologies in big data applications, thereby avoiding huge losses. Nevertheless, the variety and diversity of big data applications make it difficult to enact a theory of QA technologies to normalize them, which creates the challenge regarding a lack of awareness of QA technologies. In general, fully understanding the capabilities and limitations of QA technologies can address the specific needs of big data applications. Consequently, researchers are advised to fill this gap by deeply exploring theoretical research, considering more mature QA technologies and making use of the studies frequently applied in practice.

- Challenge 3: Lack of quantitative models and algorithms to measure the relations among big data attributes, data quality parameters and software quality attributes.

The SLR results show that big data attributes are related to the quality of software. However, big data attributes should first affect multiple data quality parameters; then, the quality of data attributes affects the quality of software. Table 9 shows our primary study on the relations among big data attributes, data quality parameters and software quality attributes. However, the change of an attribute is often accompanied by the change of multiple attributes. More detailed theories, models and algorithms are needed to precisely understand the different kinds of relations. To specify quality requirements in the context of big data applications, paper [61] presents a novel approach to address some unique requirements of engineering challenges in big data to specify quality requirements

in the context of big data applications. The approach intersects big data attributes with software quality attributes, and then it identifies the system quality requirements that apply to the intersection. Nevertheless, the approach is still in the early stages and has not been applied to the development environment of big data applications. Hence, it is still a considerable challenge and a trending research issue.

- Challenge 4: Lack of mature tools for QA technologies for big data applications.

In Section 4.3, we have summed up eight QA technologies for big data applications based on the selected 83 primary studies. Nevertheless, many authors discussed existing limitations and needed improvements. Therefore, existing technologies can solve quality problems to a certain extent. From Table 14, we can see that the 3V properties will result in software quality issues, and the corresponding technologies can partially address those problems.

Table 14. 3V properties, quality attributes and QA technologies.

3V Properties	Software Quality Attributes	Technologies
Velocity, Variety, Volume	Reliability	Specification
Velocity, Volume	Performance	Analysis
Volume	Performance, Scalability	Architectural choice
Variety, Volume	Performance, Scalability	Fault tolerance
Volume, Variety	Performance, Reliability	Verification
Variety, Velocity	Availability, Performance	Testing
Variety, Velocity	Performance	Monitoring
Variety	Performance	Fault and Failure Prediction

However, with the wide application of machine learning in the field of big data, the quality attributes of big data applications gradually appear some new non functional attributes, such as fairness and interpretability. Processing a large amount of data needs to consume more resources of the application system. The performance of big data applications is an urgent matter to be considered. Fair distribution of the resources of big data applications can greatly improve the quality of big data applications. For example, in distributed cloud computing, storage and bandwidth resources are usually limited, and such resources are usually very expensive, so collaborative users need to use resources fairly [111]. In [112], an elastic online scheduling framework is proposed to guarantee big data applications fairness. Another attribute is interpretability. Interpretability is a subjective concept on which is hard to reach a consensus. Considering the two different dimensions of semantics and complexity, when dealing with big data, researchers often pay attention to the performance indicators of big data applications, such as accuracy, but these indicators can only explain some problems, and the black box part of big data applications is still difficult to explain clearly. Due to the rapid increase of the amount of data to be processed as time goes by, the structure of big data application will gradually become more complex, which increases the difficulty of interpretation system. At this time, it is very important to study the interpretability of big data applications [113]. However, we have collected a few papers about these non functional attributes so far, and the research is still in its infancy, lacking mature tools or technologies.

In addition, although there are many mature QA tools for traditional software systems, none of the surveyed approaches discusses any mature tools that are dedicated to big data applications. Indeed, if practitioners want to apply QA technologies for big data applications today, they would have to implement their own tools, as there are no publicly available and maintained tools. This is also a very significant obstacle for the widespread use of QA technologies for big data applications in empirical research and in practice.

6. Threats to Validity

In the design of this study, several threats were encountered. Similarly to all SLR studies, a common threat to validity regards the coverage of all relevant studies. In the following, we discuss the main threats of our study and the ways we mitigated them.

External validity: In the data collection process, most of the data were collected by three researchers; this may have led to incomplete data collection, as some related articles may be missing. Although all authors have reduced the threat by determining unclear questions and discussing them together, this threat still exists. In addition, each researcher may have been biased and inflexible when he extracted the data, so at each stage of the study, we ensured that at least two other reviewers reviewed the work. Another potential threat is the consideration of studies which are only published in English. However, since the English language is the main language used for academic papers, this threat is considered to be minimal.

Internal validity: This SLR may have missed some related novel research papers. To alleviate this threat, we have searched for papers in big data-related journals/conferences/workshops. In total, 83 primary studies are selected by using the SLR. The possible threat is that QA technologies are not clearly shown for the selected primary studies. In addition, we endeavored as much as possible to extract information to analyze each article, which helps to avoid missing important information. This approach can minimize the threats as much as possible.

Construct validity: This concept relates to the validity of obtained data and the research questions. For the systematic literature review, it mainly addresses the selection of the main studies and how they represent the population of the questions. We have taken steps to reduce this threat in several ways. For example, the automatic search was performed on several electronic databases so as to avoid the potential biases.

Reliability: Reliability focuses on ensuring that the results are the same if our review would be conducted again. Different researchers who participated in the survey may be biased toward collecting and analyzing data. To solve this threat, the two researchers simultaneously extracted and analyzed data strictly according to the screening strategy, and further discussed the differences of opinion in order to enhance the objectivity of the research results. Nevertheless, the background and experience of the researchers may have produced some prejudices and introduced a certain degree of subjectivity in some cases. This threat is also related to replicating the same findings, which in turn affects the validity of the conclusion.

7. Conclusions and Future Work

A systematic literature review has been performed on QA technologies for big data applications. We did a large-scale literature search on seven electronic databases and a total of 83 papers were selected as primary studies. Regarding the research questions, we have identified that correctness, performance, availability, scalability and reliability are the focused quality attributes of big data applications; and three implementation-specific technologies, including specification, architectural choice and fault tolerance, and five process-specific technologies, including analysis, verification, testing, monitoring and fault and failure prediction have been studied for improving these quality attributes. For the focal quality attributes, the influencing factors brought by the big data attributes were identified. For the QA technologies, the strengths and limitations were compared and the empirical evidence was provided. These findings play an important role not only in the research but also in the practice of big data applications.

Although researchers have proposed these technologies to ensure quality, the research on big data quality is, however, still in its infancy, and problems regarding quality still exist in big data applications. Based on our discussions, the following topics may be part of future work:

- Considering quality attributes with big data properties together to ensure the quality of big data applications.

- Understanding and tapping into the limitations, advantages and applicable scenarios of QA technologies.
- Researching quantitative models and algorithms to measure the relations among big data properties, data quality attributes and software quality attributes.
- Developing mature tools to support QA technologies for big data applications.

Author Contributions: Conceptualization, S.J. and P.Z.; methodology, S.J. and P.Z.; investigation, Q.L.; data curation, W.C. and Q.L.; writing—original draft preparation, S.J. and W.C.; writing—review and editing, P.Z. and H.M.; supervision, P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the National Natural Science Foundation of China under grant numbers 61702159 and 61572171, and the Natural Science Foundation of Jiangsu Province under grant numbers BK20170893 and BK20191297.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dai, H.N.; Wong, R.C.W.; Wang, H.; Zheng, Z.; Vasilakos, A.V. Big data analytics for large-scale wireless networks: Challenges and opportunities. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–36. [\[CrossRef\]](#)
2. Chen, C.P.; Zhang, C.Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Inf. Sci.* **2014**, *275*, 314–347. [\[CrossRef\]](#)
3. Allam, Z.; Dhunny, Z.A. On big data, artificial intelligence and smart cities. *Cities* **2019**, *89*, 80–91. [\[CrossRef\]](#)
4. Tao, C.; Gao, J. Quality Assurance for Big Data Applications: Issues, Challenges, and Needs. In Proceedings of the Twenty-Eighth International Conference on Software Engineering and Knowledge Engineering, Redwood City, CA, USA, 1–3 July 2016.
5. Jan, B.; Farman, H.; Khan, M.; Imran, M.; Islam, I.U.; Ahmad, A.; Ali, S.; Jeon, G. Deep learning in big data analytics: A comparative study. *Comput. Electr. Eng.* **2019**, *75*, 275–287. [\[CrossRef\]](#)
6. Hilbert, M. Big Data for Development: A Review of Promises and Challenges. *Dev. Policy Rev.* **2016**, *34*, 135–174. [\[CrossRef\]](#)
7. Laranjeiro, N.; Soydemir, S.N.; Bernardino, J. A Survey on Data Quality: Classifying Poor Data. In Proceedings of the IEEE Pacific Rim International Symposium on Dependable Computing, Zhangjiajie, China, 18–20 November 2015.
8. Anagnostopoulos, I.; Zeadally, S.; Exposito, E. Handling big data: Research challenges and future directions. *J. Supercomput.* **2016**, *72*, 1494–1516. [\[CrossRef\]](#)
9. Gudivada, V.N.; Baezayates, R.; Raghavan, V.V. Big Data: Promises and Problems. *Computer* **2015**, *48*, 20–23. [\[CrossRef\]](#)
10. Montagud, S.; Abrahão, S.; Insfran, E. A systematic review of quality attributes and measures for software product lines. *Softw. Qual. J.* **2012**, *20*, 425–486. [\[CrossRef\]](#)
11. Nguyen-Duc, A.; Cruzes, D.S.; Conradi, R. The impact of global dispersion on coordination, team performance and software quality—A systematic literature review. *Inf. Soft. Technol.* **2015**, *57*, 277–294. [\[CrossRef\]](#)
12. Wang, H.; Xu, Z.; Fujita, H.; Liu, S. Towards Felicitous Decision Making: An Overview on Challenges and Trends of Big Data. *Inf. Sci.* **2016**, *367–368*, 747–765. [\[CrossRef\]](#)
13. Bagriyanik, S.; Karahoca, A. Big data in software engineering: A systematic literature review. *Glob. J. Inf. Technol. Emerg. Technol.* **2016**, *6*, 107–116. [\[CrossRef\]](#)
14. Schulmeyer, G.G.; McManus, J.I. *Handbook of Software Quality Assurance*; Van Nostrand Reinhold Co.: New York, NY, USA, 1992.
15. Gao, J.; Xie, C.; Tao, C. Big Data Validation and Quality Assurance —Issues, Challenges, and Needs. In Proceedings of the 2016 IEEE Symposium on Service-Oriented System Engineering, Oxford, UK, 29 March–2 April 2016; pp. 433–441.
16. Lai, S.T.; Leu, F.Y. Data Preprocessing Quality Management Procedure for Improving Big Data Applications Efficiency and Practicality; In Proceedings of the International Conference on Broadband and Wireless Computing, Communication and Applications, Asan, Korea, 5–7 November 2016; pp. 731–738.

17. Garg, N.; Singla, S.; Jangra, S. Challenges and techniques for testing of big data. *Procedia Comput. Sci.* **2016**, *85*, 940–948. [CrossRef]
18. Zhou, H.; Lou, J.G.; Zhang, H.; Lin, H.; Lin, H.; Qin, T. An Empirical Study on Quality Issues of Production Big Data Platform. In Proceedings of the IEEE/ACM IEEE International Conference on Software Engineering, Florence, Italy, 16–24 May 2015; pp. 17–26.
19. Juddoo, S. Overview of data quality challenges in the context of Big Data. In Proceedings of the International Conference on Computing, Communication and Security, Pamplemousses, Mauritius, 4–5 December 2016.
20. Zhang, P.; Zhou, X.; Gao, J.; Tao, C. A survey on quality assurance techniques for big data applications. In Proceedings of the IEEE BigDataService 2017—International Workshop on Quality Assurance and Validation for Big Data Applications, San Francisco, CA, USA, 6–9 April 2017.
21. Ge, M.; Dohnal, V. Quality Management in Big Data. *Informatics* **2018**, *5*, 19. [CrossRef]
22. Liu, J.; Li, J.; Li, W.; Wu, J. Rethinking big data: A review on the data quality and usage issues. *ISRRS J. Photogramm. Remote Sens.* **2016**, *115*, 134–142. [CrossRef]
23. Ledmi, A.; Bendjenna, H.; Hemam, M.S. Fault Tolerance in Distributed Systems: A Survey. In Proceedings of the 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa, Algeria, 24–25 October 2018; pp. 1–5.
24. Niedermaier, S.; Koetter, F.; Freymann, A.; Wagner, S. On Observability and Monitoring of Distributed Systems: An Industry Interview Study. In Proceedings of the International Conference on Service-Oriented Computing, Toulouse, France, 28–31 October 2019; pp. 36–52.
25. Kitchenham, B.; Pretorius, R.; Budgen, D.; Brereton, O.P.; Turner, M.; Niazi, M.; Linkman, S. Systematic literature reviews in software engineering: A tertiary study. *Inf. Softw. Technol.* **2009**, *51*, 7–15. [CrossRef]
26. Kitchenham, B. Procedures for Performing Systematic Reviews. *Keele* **2004**, *33*, 1–26.
27. Basili, V.R.; Caldiera, G.; Rombach, H.D. The goal question metric approach. In *Encyclopedia of Software Engineering*; Wiley: Hoboken, NJ, USA, 1994.
28. Kitchenham, B.A.; Dyba, T.; Jorgensen, M. Evidence-Based Software Engineering. In Proceedings of the International Conference on Software Engineering, ICSE, Edinburgh, UK, 23–28 May 2004; pp. 273–281.
29. Zhang, H.; Ali Babar, M. On searching relevant studies in software engineering. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, Keele, UK, 12–13 April 2010.
30. Goodman, L.A. Snowball Sampling. *Ann. Math. Stat.* **1961**, *32*, 148–170. [CrossRef]
31. Rosà, A.; Chen, L.Y.; Binder, W. Failure analysis and prediction for big-data systems. *IEEE Trans. Serv. Comput.* **2017**, *10*, 984–998. [CrossRef]
32. Cao, R.; Gao, J. Research on reliability evaluation of big data system. In Proceedings of the 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 20–22 April 2018; pp. 261–265.
33. Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering. 2007. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.471&rep=rep1&type=pdf> (accessed on 12 November 2020).
34. Aggarwal, A. Identification of quality parameters associated with 3V's of Big Data. In Proceedings of the International Conference on Computing for Sustainable Global Development, New Delhi, India, 16–18 March 2016.
35. Fasel, D. Potentials of big data for governmental services. In Proceedings of the First International Conference on Edemocracy and Egovernment, Quito, Ecuador, 24–25 April 2014.
36. Becker, D.; King, T.D.; McMullen, B. Big data, big data quality problem. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2644–2653.
37. Clarke, R. Big data, big risks. *Inf. Syst. J.* **2016**, *26*, 77–90. [CrossRef]
38. Liu, Z. Research of performance test technology for big data applications. In Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA), Hailar, China, 28–30 July 2014; pp. 53–58.
39. Huang, J.; Niu, L.; Zhan, J.; Peng, X.; Bai, J.; Cheng, S. Technical aspects and case study of big data based condition monitoring of power apparatuses. In Proceedings of the 2014 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Hong Kong, China, 7–10 December 2015; pp. 1–4.

40. Li, N.; Escalona, A.; Guo, Y.; Offutt, J. A scalable big data test framework. In Proceedings of the 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), Graz, Austria, 13–17 April 2015; pp. 1–2.
41. Scavuzzo, M.; Tamburri, D.A.; Di Nitto, E. Providing big data applications with fault-tolerant data migration across heterogeneous NoSQL databases. In Proceedings of the 2016 IEEE/ACM 2nd International Workshop on Big Data Software Engineering (BIGDSE), Austin, TX, USA, 16 May 2016; pp. 26–32.
42. Xu, C.; Holzemer, M.; Kaul, M.; Markl, V. Efficient fault-tolerance for iterative graph processing on distributed dataflow systems. In Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland, 16–20 May 2016; pp. 613–624.
43. Ding, J.; Hu, X.H.; Gudivada, V. A Machine Learning Based Framework for Verification and Validation of Massive Scale Image Data. *IEEE Trans. Big Data* **2017**, *1*. [[CrossRef](#)]
44. Lundberg, L.; Grahn, H.; Ilie, D.; Melander, C. Cache support in a high performance fault-tolerant distributed storage system for cloud and big data. In Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, Hyderabad, India, 25–29 May 2015; pp. 537–546.
45. Fredericks, E.M.; Hariri, R.H. Extending search-based software testing techniques to big data applications. In Proceedings of the 2016 IEEE/ACM 9th International Workshop on Search-Based Software Testing (SBST), Austin, TX, USA, 16–17 May 2017; pp. 41–42.
46. Lin, J.; Liang, F.; Lu, X.; Zha, L.; Xu, Z. Modeling and designing fault-tolerance mechanisms for mpi-based mapreduce data computing framework. In Proceedings of the 2015 IEEE First International Conference on Big Data Computing Service and Applications, Redwood City, CA, USA, 30 March–2 April 2015; pp. 176–183.
47. Morán, J.; Bertolino, A.; de la Riva, C.; Tuya, J. Automatic testing of design faults in mapreduce applications. *IEEE Trans. Reliab.* **2018**, *67*, 717–732. [[CrossRef](#)]
48. Zhao, X.; Gao, X. An ai software test method based on scene deductive approach. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 14–20.
49. Jiang, C.; Huang, S.; Hui, Z.W. Metamorphic Testing of Image Region Growth Programs in Image Processing Applications. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal 16–20 July 2018; pp. 70–72.
50. Shi, M.; Yuan, R. Mad: A monitor system for big data applications. In Proceedings of the International Conference on Intelligent Science and Big Data Engineering, Suzhou, China, 14–16 June 2015; pp. 308–315.
51. Yang, Y.; Ai, J.; Wang, F. Defect prediction based on the characteristics of multilayer structure of software network. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 27–34.
52. Malhotra, R. Comparative analysis of statistical and machine learning methods for predicting faulty modules. *Appl. Soft Comput.* **2014**, *21*, 286–297. [[CrossRef](#)]
53. Akash, G.; Lee, O.T.; Kumar, S.M.; Chandran, P.; Cuzzocrea, A. Rapid: A fast data update protocol in erasure coded storage systems for big data. In Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, Spain, 14–17 May 2017; pp. 890–897.
54. Shi, G.; Wang, H. Research on big data real-time public opinion monitoring under the double cloud architecture. In Proceedings of the 2016 IEEE Second International Conference on Multimedia Big Data (BigMM), Taipei, Taiwan, 20–22 April 2016; pp. 416–419.
55. Liu, X.; Fan, X.; Li, J. A Novel Parallel Architecture with Fault-Tolerance for Joining Bi-directional Data Streams in Cloud. In Proceedings of the 2013 International Conference on Cloud Computing and Big Data, Fuzhou, China, 16–19 December 2013; pp. 30–37.
56. Iuhasz, G.; Dragan, I. An overview of monitoring tools for big data and cloud applications. In Proceedings of the 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNAS), Timisoara, Romania, 21–24 September 2015; pp. 363–366.
57. Ganguly, S.; Consul, A.; Khan, A.; Bussone, B.; Richards, J.; Miguel, A. A practical approach to hard disk failure prediction in cloud platforms: Big data model for failure management in datacenters. In Proceedings of the 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 29 March–1 April 2016; pp. 105–116.
58. Wang, Y.; Shen, Y.; Wang, H.; Cao, J.; Jiang, X. Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications. *IEEE Trans. Big Data* **2016**, *4*, 418–431. [[CrossRef](#)]

59. Puthal, D.; Nepal, S.; Ranjan, R.; Chen, J. DLSeF: A dynamic key-length-based efficient real-time security verification model for big data stream. *ACM Trans. Embed. Comput. Syst. (TECS)* **2016**, *16*, 1–24. [[CrossRef](#)]
60. Villalpando, L.E.B.; April, A.; Abran, A. Performance analysis model for big data applications in cloud computing. *J. Cloud Comput.* **2014**, *3*, 19. [[CrossRef](#)]
61. Noorwali, I.; Arruda, D.; Madhavji, N.H. Understanding quality requirements in the context of big data systems. In Proceedings of the 2nd International Workshop on BIG Data Software Engineering, Austin, TX, USA, 16 May 2016; pp. 76–79.
62. Liu, C.; Yang, C.; Zhang, X.; Chen, J. External integrity verification for outsourced big data in cloud and IoT: A big picture. *Future Gener. Comput. Syst.* **2015**, *49*, 58–67. [[CrossRef](#)]
63. Zhang, L. A framework to specify big data driven complex cyber physical control systems. In Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA), Hailar, China, 28–30 July 2014; pp. 548–553.
64. Eichelberger, H. Flexible System-Level Monitoring of Heterogeneous Big Data Streaming Systems. In Proceedings of the 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Prague, Czech Republic, 29–31 August 2018; pp. 289–292.
65. Xiaorong, F.; Shizhun, J.; Songtao, M. The research on industrial big data information security risks. In Proceedings of the 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China, 9–12 March 2018; pp. 19–23.
66. Andreolini, M.; Pietri, M.; Tosi, S.; Balboni, A. Monitoring large cloud-based systems. In Proceedings of the 4th International Conference on Cloud Computing and Services Science, CLOSER 2014, Barcelona, Spain, 3–5 April 2014; pp. 341–351.
67. Shen, C.; Tong, W.; Choo, K.K.R.; Kausar, S. Performance prediction of parallel computing models to analyze cloud-based big data applications. *Clust. Comput.* **2017**, *21*, 1439–1454. [[CrossRef](#)]
68. Lin, H.Y.; Yang, S.Y. A cloud-based energy data mining information agent system based on big data analysis technology. *Microelectron. Reliab.* **2019**, *97*, 66–78. [[CrossRef](#)]
69. Tsui, K.L.; Zhao, Y.; Wang, D. Big data opportunities: System health monitoring and management. *IEEE Access* **2019**, *7*, 68853–68867. [[CrossRef](#)]
70. Rao, T.R.; Mitra, P.; Bhatt, R.; Goswami, A. The big data system, components, tools, and technologies: A survey. *Knowl. Inf. Syst.* **2019**, 1–81. [[CrossRef](#)]
71. Alhamazani, K.; Ranjan, R.; Jayaraman, P.P.; Mitra, K.; Wang, M.; Huang, Z.G.; Wang, L.; Rabhi, F. Real-time qos monitoring for cloud-based big data analytics applications in mobile environments. In Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management, Brisbane, QLD, Australia, 14–18 July 2014; Volume 1, pp. 337–340.
72. Alodib, M.; Malik, Z. A Big Data approach to enhance the integration of Access Control Policies for Web Services. In Proceedings of the 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), Las Vegas, NV, USA, 28 June–1 July 2015; pp. 41–46.
73. Jhawar, R.; Piuri, V. Fault tolerance and resilience in cloud computing environments. In *Cyber Security and IT Infrastructure Protection*; Syngress: Boston, MA, USA, 2014; pp. 165–181.
74. Rajbhoj, A.; Kulkarni, V.; Bellarykar, N. Early experience with model-driven development of mapreduce based big data application. In Proceedings of the 2014 21st Asia-Pacific Software Engineering Conference, Jeju, Korea, 1–4 December 2014; Volume 1, pp. 94–97.
75. Ficco, M.; Palmieri, F.; Castiglione, A. Modeling security requirements for cloud-based system development. *Concurr. Comput. Pract. Exp.* **2015**, *27*, 2107–2124. [[CrossRef](#)]
76. Nagy, C. Static analysis of data-intensive applications. In Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering, Genova, Italy, 5–8 March 2013; pp. 435–438.
77. Tuma, K.; Calikli, G.; Scandariato, R. Threat analysis of software systems: A systematic literature review. *J. Syst. Softw.* **2018**, *144*, 275–294. [[CrossRef](#)]
78. Etani, N. Database application model and its service for drug discovery in Model-driven architecture. *J. Big Data* **2015**, *2*, 16. [[CrossRef](#)]
79. Tolosana-Calasanz, R.; Banares, J.A.; Colom, J.M. Model-driven development of data intensive applications over cloud resources. *Future Gener. Comput. Syst.* **2018**, *87*, 888–909. [[CrossRef](#)]
80. Wang, H.; Chen, H.; Du, Z.; Hu, F. Betl: Mapreduce checkpoint tactics beneath the task level. *IEEE Trans. Serv. Comput.* **2017**, *9*, 84–95. [[CrossRef](#)]

81. Camilli, M. Formal verification problems in a big data world: towards a mighty synergy. In Proceedings of the Companion Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India, 31 May–7 June 2014; pp. 638–641.
82. Nambiar, R.; Poess, M.; Dey, A.; Cao, P.; Magdon-Ismael, T.; Bond, A. Introducing TPCx-HS: The first industry standard for benchmarking big data systems. In Proceedings of the Technology Conference on Performance Evaluation and Benchmarking, Hangzhou, China, 11 August 2014; pp. 1–12.
83. Syer, M.D.; Shang, W.; Jiang, Z.M.; Hassan, A.E. Continuous validation of performance test workloads. *Autom. Softw. Eng.* **2017**, *24*, 189–231. [[CrossRef](#)]
84. Zhang, W.; Liu, W.; Wei, B. Software system testing method based on formal model. In Proceedings of the 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 28–30 April 2017; pp. 410–415.
85. Malhotra, R.; Bahl, L.; Sehgal, S.; Priya, P. Empirical comparison of machine learning algorithms for bug prediction in open source software. In Proceedings of the 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, India, 23–25 March 2017; pp. 40–45.
86. Fahmideh, M.; Beydoun, G. Big data analytics architecture design—An application in manufacturing systems. *Comput. Ind. Eng.* **2019**, *128*, 948–963. [[CrossRef](#)]
87. Gonzalez-Aparicio, M.T.; Younas, M.; Tuya, J.; Casado, R. Evaluation of ACE properties of traditional SQL and NoSQL big data systems. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 1988–1995.
88. Zamfir, V.A.; Carabas, M.; Carabas, C.; Tapus, N. Systems monitoring and big data analysis using the elasticsearch system. In Proceedings of the 2019 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 28–30 May 2019; pp. 188–193.
89. Zou, H.; Yu, Y.; Tang, W.; Chen, H.W.M. FlexAnalytics: A flexible data analytics framework for big data applications with I/O performance improvement. *Big Data Res.* **2014**, *1*, 4–13. [[CrossRef](#)]
90. Bu, Y.; Borkar, V.; Xu, G.; Carey, M.J. A bloat-aware design for big data applications. In Proceedings of the 2013 international symposium on memory management, Seattle, WA, USA, 20 June 2013; pp. 119–130.
91. Andreolini, M.; Colajanni, M.; Pietri, M.; Tosi, S. Adaptive, scalable and reliable monitoring of big data on clouds. *J. Parallel Distrib. Comput.* **2015**, *79*, 67–79. [[CrossRef](#)]
92. Osvaldo, S.S., Jr.; Lopes, D.; Silva, A.C.; Abdelouahab, Z. Developing software systems to Big Data platform based on MapReduce model: An approach based on Model Driven Engineering. *Inf. Softw. Technol.* **2017**, *92*, 30–48. [[CrossRef](#)]
93. Enes, J.; Exposito, R.R.; Tourino, J. BDWatchdog: Real-time monitoring and profiling of Big Data applications and frameworks. *Future Gener. Comput. Syst.* **2018**, *87*, 420–437. [[CrossRef](#)]
94. Zareian, S.; Fokaefs, M.; Khazaei, H.; Litoiu, M.; Zhang, X. A big data framework for cloud monitoring. In Proceedings of the 2nd International Workshop on BIG Data Software Engineering, Austin, TX, USA, 16 May 2016; pp. 58–64.
95. Casale, G.; Ardagna, D.; Artac, M.; Barbier, F.; Di Nitto, E.; Henry, A.; Iuhasz, G.; Joubert, C.; Merseguer, J.; Munteanu, V.I. DICE: Quality-driven development of data-intensive cloud applications. In Proceedings of the 2015 IEEE/ACM 7th International Workshop on Modeling in Software Engineering, Florence, Italy, 16–17 May 2015; pp. 78–83.
96. Fang, K.; Li, X.; Hao, J.; Feng, Z. Formal modeling and verification of security protocols on cloud computing systems based on UML 2.3. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 852–859.
97. Klein, J.; Gorton, I.; Alhmoud, L.; Gao, J.; Gemici, C.; Kapoor, R.; Nair, P.; Saravagi, V. Model-driven observability for big data storage. In Proceedings of the 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), Venice, Italy, 5–8 April 2016; pp. 134–139.
98. Amato, F.; Moscato, F. Model transformations of mapreduce design patterns for automatic development and verification. *J. Parallel Distrib. Comput.* **2016**, *110*, 52–59. [[CrossRef](#)]
99. Majd, A.; Troubitsyna, E. Data-driven approach to ensuring fault tolerance and efficiency of swarm systems. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 4792–4794.
100. Li, H.; Wu, J.; Jiang, Z.; Li, X.; Wei, X. Minimum backups for stream processing with recovery latency guarantees. *IEEE Trans. Reliab.* **2017**, *66*, 783–794. [[CrossRef](#)]

101. Shang, W.; Jiang, Z.M.; Hemmati, H.; Adams, B.; Hassan, A.E.; Martin, P. Assisting developers of big data analytics applications when deploying on hadoop clouds. In Proceedings of the 2013 35th International Conference on Software Engineering (ICSE), San Francisco, CA, USA, 18–26 May 2013; pp. 402–411.
102. Guo, C.; Zhu, S.; Wang, T.; Wang, H. FeT: Hybrid Cloud-Based Mobile Bank Application Testing. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 21–26.
103. Wang, J.; Ren, D. Research on Software Testing Technology Under the Background of Big Data. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; pp. 2679–2682.
104. Shafiq, M.O.; Fekri, M.; Ibrahim, R. MapReduce based classification for fault detection in big data applications. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 637–642.
105. Zhang, K.; Xu, J.; Min, M.R.; Jiang, G.; Pelechrinis, K.; Zhang, H. Automated IT system failure prediction: A deep learning approach. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2017; pp. 1291–1300.
106. Wu, X.; Du, Z.; Dai, S.; Liu, Y. The Fault Tolerance of Big Data Systems. In Proceedings of the International Workshop on Management of Information, Processes and Cooperation, Hangzhou, China, 27 February 2017; pp. 65–74.
107. Hazen, B.T.; Boone, C.A.; Ezell, J.D.; Jones-Farmer, L.A. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *Int. J. Prod. Econ.* **2014**, *154*, 72–80. [[CrossRef](#)]
108. Guan, Q.; Zhang, Z.; Fu, S. Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems. *J. Commun.* **2012**, *7*, 52–61. [[CrossRef](#)]
109. Yin, J.; Zhao, D. Data confidentiality challenges in big data applications. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2886–2888.
110. Al-Najran, N.; Dahanayake, A. A requirements specification framework for big data collection and capture. In Proceedings of the East European Conference on Advances in Databases and Information Systems, Poitiers, France, 8–11 September 2015; pp. 12–19.
111. Xia, Q.; Xu, Z.; Liang, W.; Zomaya, A.Y. Collaboration-and fairness-aware big data management in distributed clouds. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *27*, 1941–1953. [[CrossRef](#)]
112. Sun, D.; Yan, H.; Gao, S.; Liu, X.; Buyya, R. Rethinking elastic online scheduling of big data streaming applications over high-velocity continuous data streams. *J. Supercomput.* **2019**, *74*, 615–636. [[CrossRef](#)]
113. Barsacchi, M.; Bechini, A.; Ducange, P.; Marcelloni, F. Optimizing partition granularity, membership function parameters, and rule bases of fuzzy classifiers for big data by a multi-objective evolutionary approach. *Cogn. Comput.* **2019**, *11*, 367–387. [[CrossRef](#)]
114. Sneed, H.M.; Erdoes, K. Testing big data (Assuring the quality of large databases). In Proceedings of the 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Graz, Austria, 13–17 April 2015; pp. 1–6.
115. Dai, D.; Chen, Y.; Kimpe, D.; Ross, R. Provenance-based object storage prediction scheme for scientific big data applications. In Proceedings of the 2014 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 27–30 October 2015; pp. 271–280.
116. Rabl, T.; Sadoghi, M.; Jacobsen, H.A.; Gomez-Villamor, S.; Munteș-Mulero, V.; Mankowskii, S. Solving big data challenges for enterprise application performance management. *arXiv* **2012**, arXiv:1208.4167.
117. Hussain, M.; Al-Mourad, M.B.; Mathew, S.S. Collect, Scope, and Verify Big Data—A Framework for Institution Accreditation. In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops, Crans-Montana, Switzerland, 23–25 March 2016.
118. Elkafrawy, P.M.; Sauber, A.M.; Hafez, M.M. HDFSX: Big data Distributed File System with small files support. In Proceedings of the Computer Engineering Conference, Cairo, Egypt, 28–29 December 2017; pp. 131–135.
119. Radha, K.R.; Karthik, S. Efficient Handling of Big Data Volume Using Heterogeneous Distributed File Systems. *Int. J. Comput. Trends Technol.* **2014**, *15*. [[CrossRef](#)]

120. Ke, H.; Li, P.; Guo, S.; Guo, M. On Traffic-Aware Partition and Aggregation in MapReduce for Big Data Applications. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 818–828. [[CrossRef](#)]
121. Dobre, C.; Xhafa, F. Parallel Programming Paradigms and Frameworks in Big Data Era. *Int. J. Parallel Program.* **2014**, *42*, 710–738. [[CrossRef](#)]
122. Reniers, V.; Landuyt, D.V.; Rafique, A.; Joosen, W. On the State of NoSQL Benchmarks. In Proceedings of the ACM/SPEC on International Conference on PERFORMANCE Engineering Companion, L'Aquila, Italy, 22–26 April 2017; pp. 107–112.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).