Kolajo *et al. J Big Data* (2019) 6:47
https://doi.org/10.1186/s40537-019-0210-7

Journal of Big Data

SURVEY PAPER Open Access

Big data stream analysis: a systematic literature review

Taiwo Kolajo[1,2*], Olawande Daramola[3] and Ayodele Adebiyi[1,4]

*Correspondence:
taiwo.kolajo@stu.cu.edu.ng;
taiwo.kolajo@fulokoja.edu.ng
[1] Department of Computer and Information Sciences, Covenant University, Ota, Nigeria
Full list of author information is available at the end of the article

Abstract

Recently, big data streams have become ubiquitous due to the fact that a number of applications generate a huge amount of data at a great velocity. This made it difficult for existing data mining tools, technologies, methods, and techniques to be applied directly on big data streams due to the inherent dynamic characteristics of big data. In this paper, a systematic review of big data streams analysis which employed a rigorous and methodical approach to look at the trends of big data stream tools and technologies as well as methods and techniques employed in analysing big data streams. It provides a global view of big data stream tools and technologies and its comparisons. Three major databases, Scopus, ScienceDirect and EBSCO, which indexes journals and conferences that are promoted by entities such as IEEE, ACM, SpringerLink, and Elsevier were explored as data sources. Out of the initial 2295 papers that resulted from the first search string, 47 papers were found to be relevant to our research questions after implementing the inclusion and exclusion criteria. The study found that scalability, privacy and load balancing issues as well as empirical analysis of big data streams and technologies are still open for further research efforts. We also found that although, significant research efforts have been directed to real-time analysis of big data stream not much attention has been given to the preprocessing stage of big data streams. Only a few big data streaming tools and technologies can do all of the batch, streaming, and iterative jobs; there seems to be no big data tool and technology that offers all the key features required for now and standard benchmark dataset for big data streaming analytics has not been widely adopted. In conclusion, it was recommended that research efforts should be geared towards developing scalable frameworks and algorithms that will accommodate data stream computing mode, effective resource allocation strategy and parallelization issues to cope with the ever-growing size and complexity of data.

Keywords: Big data stream analysis, Stream computing, Big data streaming tools and technologies

Introduction

Advances in information technology have facilitated large volume, high-velocity of data, and the ability to store data continuously leading to several computational challenges. Due to the nature of big data in terms of volume, velocity, variety, variability, veracity, volatility, and value [1] that are being generated recently, big data computing is a new trend for future computing.

Big data computing can be generally categorized into two types based on the processing requirements, which are big data batch computing and big data stream computing

[2]. Big data batch processing is not sufficient when it comes to analysing real-time application scenarios. Most of the data generated in a real-time data stream need real-time data analysis. In addition, the output must be generated with low-latency and any incoming data must be reflected in the newly generated output within seconds. This necessitates big data stream analysis [3].

The demand for stream processing is increasing. The reason being not only that huge volume of data need to be processed but that data must be speedily processed so that organisations or businesses can react to changing conditions in real-time.

This paper presents a systematic review of big data stream analysis. The purpose is to present an overview of research works, findings, as well as implications for research and practice. This is necessary to (1) provide an update about the state of research, (2) identify areas that are well researched, (3) showcase areas that are lacking and need further research, and (4) build a common understanding of the challenges that exist for the benefit of the scientific community.

The rest of the paper is organized as follows: "Background and related work" section provides information on stream computing and big data stream analysis and the key issues involved in it and presents a review on big data streaming analytics. In "Research method" section, the adopted research methodology is discussed, while "Result" section presents the findings of the study. "Discussion" section presents a detailed evaluation performed on big data stream analysis, "Limitation of the review" section highlights the limitations of the study, while "Conclusion and further work" concludes the paper.

Background and related work

Stream computing

Stream computing refers to the processing of massive amount of data generated at high-velocity from multiple sources with low latency in real-time. It is a new paradigm necessitated because of new sources of data generating scenarios which include ubiquity of location services, mobile devices, and sensor pervasiveness [4]. It can be applied to the high-velocity flow of data from real-time sources such as the Internet of Things, Sensors, market data, mobile, and clickstream.

The fundamental assumption of this paradigm is that the potential value of data lies in its freshness. As a result, data are analysed as soon as they arrive in a stream to produce result as opposed to what obtains in batch computing where data are first stored before they are analysed. There is a crucial need for parallel architectures and scalable computing platforms [5]. With stream computing, organisations can analyse and respond in real-time to rapidly changing data. Streaming processing frameworks include Storm, S4, Kafka, and Spark [6–8]. The real contrasts between the batch processing and the stream processing paradigms are outlined in Table 1.

Incorporating streaming data into decision-making process necessitates a programming paradigm called stream computing. With stream computing, fairly static questions can be evaluated on data in motion (i.e. real-time data) continuously [9].

Big data stream analysis

The essence of big data streaming analytics is the need to analyse and respond to real-time streaming data using continuous queries so that it is possible to continuously

Table 1 Comparison between batch processing and streaming processing [82]

Dimension	Batch processing	Streaming processing
Input	Data chunks	Stream of new data or updates
Data size	Known and finite	Infinite or unknown in advance
Hardware	Multiple CPUs	Typical single limited amount of memory
Storage	Store	Not store or store non-trivial portion in memory
Processing	Processed in multiple rounds	A single or few passes over data
Time	Much longer	A few seconds or even milliseconds
Applications	Widely adopted in almost every domain	Web mining, traffic monitoring, sensor networks

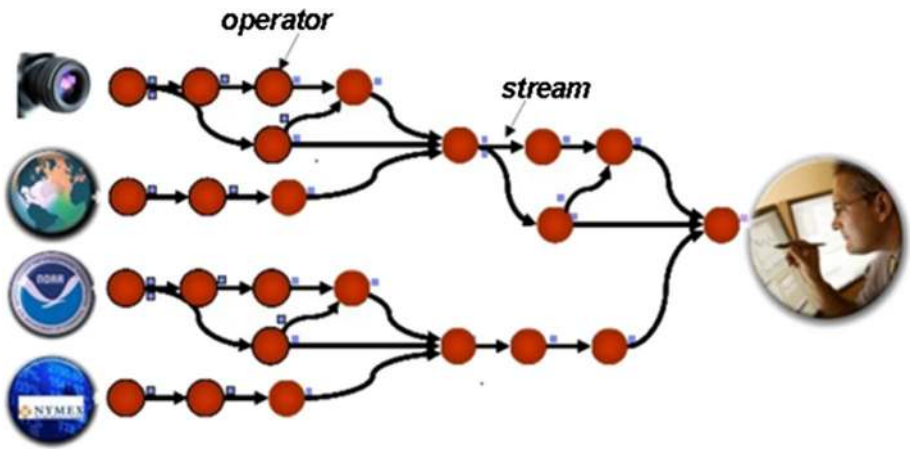

Fig. 1 Data flow graph of a stream processor. The figure shows how applications (made up of operations and interconnected streams) are represented as data flow graph in a stream processor [10]

perform analysis on the fly within the stream. Stream processing solutions must be able to handle a real-time, high volume of data from diverse sources putting into consideration availability, scalability and fault tolerance. Big data stream analysis involves assimilation of data as an infinite tuple, analysis and production of actionable results usually in a form of stream [10].

In a stream processor, applications are represented as data flow graph made up of operations and interconnected streams as depicted in Fig. 1. In a streaming analytics system, application comes in a form of continuous queries, data are ingested continuously, analysed and correlated, and stream of results are generated. Streaming analytic applications is usually a set of operators connected by streams. Streaming analytics systems must be able to identify new information, incrementally build models and access whether the new incoming data deviate from model predictions [9].

The idea of streaming analytics is that each of the received data tuples is processed in the data processing node. Such processing includes removing duplicates, filling missing data, data normalization, parsing, feature extraction, which are typically done in a single pass due to the high data rates of external feeds. When a new tuple arrives, this node is triggered, and it expels tuples older than the time specified in the sliding window (sliding window is a typical example of windows used in stream computing which keeps only the latest tuples up to the time specified in the windows). A window

is referred to as a logical container for data tuples received. It defines how frequently data is refreshed in the container as well as when data processing is triggered [4].

Key issues in big data stream analysis

Big data stream analysis is relevant when there is a need to obtain useful knowledge from current happenings in an efficient and speedy manner in order to enable organisations to quickly react to problems, or detect new trends which can help improve their performance. However, there are some challenges such as scalability, integration, fault-tolerance, timeliness, consistency, heterogeneity and incompleteness, load balancing, privacy issues, and accuracy [3, 11–18] which arises from the nature of big data streams that must be dealt with.

Scalability

One of the main challenges in big data streaming analysis is the issue of scalability. The big data stream is experiencing exponential growth in a way much faster than computer resources. The processors follow Moore's law, but the size of data is exploding. Therefore, research efforts should be geared towards developing scalable frameworks and algorithms that will accommodate data stream computing mode, effective resource allocation strategy and parallelization issues to cope with the ever-growing size and complexity of data.

Integration

Building a distributed system where each node has a view of the data flow, that is, every node performing analysis with a small number of sources, then aggregating these views to build a global view is non-trivial. An integration technique should be designed to enable efficient operations across different datasets.

Fault-tolerance

High fault-tolerance is required in life-critical systems. As data is real-time and infinite in big data stream computing environments, a good scalable high fault-tolerance strategy is required that allows an application to continue working despite component failure without interruption.

Timeliness

Time is of the essence for time-sensitive processes such as mitigating security threats, thwarting fraud, or responding to a natural disaster. There is a need for scalable architectures or platforms that will enable continuous processing of data streams which can be used to maximize the timeliness of data. The main challenge is implementing a distributed architecture that will aggregate local views of data into global view with minimal latency between communicating nodes.

Consistency

Achieving high consistency (i.e. stability) in big data stream computing environments is non-trivial as it is difficult to determine which data are needed and which nodes should be consistent. Hence a good system structure is required.

Heterogeneity and incompleteness

Big data streams are heterogeneous in structure, organisations, semantics, accessibility and granularity. The challenge here is how to handle an always ever-increasing data, extract meaningful content out of it, aggregate and correlate streaming data from multiple sources in real-time. A competent data presentation should be designed to reflect the structure, diversity and hierarchy of the streaming data.

Load balancing

A big data stream computing system is expected to be self-adaptive to data streams changes and avoid load shedding. This is challenging as dedicating resources to cover peak loads 24/7 is impossible and load shedding is not feasible when the variance between the average load and the peak load is high. As a result, a distributing environment that automatically streams partial data streams to a global centre when local resources become insufficient is required.

High throughput

Decision with respect to identifying the sub-graph that needs replication, how many replicas are needed and the portion of the data stream to assign to each replica is an issue in big data stream computing environment. There is a need for good multiple instances replication if high throughput is to be achieved.

Privacy

Big data stream analytics created opportunities for analyzing a huge amount of data in real-time but also created a big threat to individual privacy. According to the International Data Cooperation (IDC), not more than half of the entire information that needs protection is effectively protected. The main challenge is proposing techniques for protecting a big data stream dataset before its analysis.

Accuracy

One of the main objectives of big data stream analysis is to develop effective techniques that can accurately predict future observations. However, as a result of inherent characteristics of big data such as volume, velocity, variety, variability, veracity, volatility, and value, big data analysis strongly constrain processing algorithms spatio-temporally and hence stream-specific requirements must be taken into consideration to ensure high accuracy.

Related work

This section discusses some of the previous research efforts that relate to big data streaming analytics.

The work of [13] presented a review of various tools, technologies and methods for big data analytics by categorizing big data analytics literature according to their research focus. This paper is different in that it presents a systematic literature review that focused on big data "streaming" analytics.

Authors in [19] presented a systematic review of big data analytics in e-commerce. The study explored characteristics, definitions, business values, types and challenges of big data analytics in the e-commerce landscape. Likewise, [20] conducted a study that is centred on big data analytics in technology and organisational resource management specifically focusing on reviews that present big data challenges and big data analytics methods. Although they are systematic reviews, the focus is not, particularly on big data streaming.

Authors in [21] presented the status of empirical research and application areas in big data by employing a systematic mapping method. In the same vein, authors in [22] also conducted a survey on big data technologies and machine learning algorithms with a particular focus on anomaly detection. A systematic review of literature which aims to determine the scope, application, and challenges of big data analytics in healthcare was presented by [23]. The work of [2] presented a review of four big data streaming tools and technologies. While the study conducted in this paper provided a comprehensive review of not only big data streaming tools and technologies but also methods and techniques employed in analyzing big data streams. In addition, authors [2] did not provide a clear explanation of the methodical approach for selecting the reviewed papers.

Research method

The study was grounded in a systematic literature review of tools and technologies with methods and techniques used in analysing big data streams by adopting [24, 25] as models.

Research question

The study tries to answer the following research questions:

Research Question 1: What are the tools and technologies employed for big data stream analysis?
Research Question 2: What methods and techniques are used in analysing big data streams?
Research Question 3: What do these tools and technologies have in common and their differences in terms of concept, purpose and capabilities?
Research Question 4: What are the limitations and strengths of these tools and technologies?
Research Question 5: What are the evaluation techniques or benchmarks used for evaluating big data streaming tools and technology?

Search string

Creating a good search string requires structuring in terms of population, comparison, intervention and outcome [24]. Relevant publications were identified by forming a search string that combined keywords driven by the research questions earlier stated. The searches were conducted by employing three standard database indexes, which are Scopus, Science Direct and EBSCOhost. The search string is “big data stream analysis” OR “big data stream technologies” OR “big data stream framework” OR “big data stream algorithms” OR “big data stream analysis tools” OR “big data stream processing” OR “big

data stream analysis reviews" OR "big data stream literature review" OR "big data stream analytics".

Data sources

As research becomes increasingly interdisciplinary, global and collaborative, it is expedient to select from rich and standard databases. The databases consulted are as follows:

i. Scopus[1]: Scopus is a bibliographic database containing abstracts and citations for academic journal articles launched in 2004. It covers nearly 36,377 titles from over 11,678 publishers of which 34,346 are peer-reviewed journals, delivering a comprehensive overview of the world's research output in the scientific, technical, medical, and social sciences (including arts and humanities). It is the largest abstract and citation database of peer-reviewed literature.
ii. ScienceDirect[2]: ScienceDirect is Elsevier's leading information solution for researchers, students, teachers, information professionals and healthcare professionals. It provides both subscription-based and open access-based to a large database combining authoritative, full-text scientific, technical and health publications with smart intuitive functionality. It covers over 14 million publications from over 3800 journals and more than 35,000 books. The journals are grouped into four categories: Life Sciences, Physical Sciences and Engineering, Health Sciences, and Social Sciences and Humanities.
iii. EBSCOhost[3]: EBSCOhost covers a wide range of bibliographic and full-text databases for researchers, providing electronic journal service available to both corporate and academic researchers. It has a total of 16,711 journals and magazine indexed and abstracted of which 14,914 are peer-reviewed; more than 900,000 high-quality e-books and titles and over 60,000 audiobooks from more than 1500 major academic publishers.
iv. ResearchGate[4]: A free online professional network for scientists and researchers to ask and answer questions, share papers and find collaborators. It covers over 100 million publications from over 11 million researchers. ResearchGate was used as a secondary source where the authors could not access some papers due to lack of subscription.

Data retrieval

The search was conducted in Scopus, ScienceDirect and EBSCOhost since most of the high impact journals and conferences are indexed in these set of rich databases. Boolean 'OR' was used in combining the nine (9) search strings. A total of 2295 articles from the three databases were retrieved as shown in Table 2.

[1] http://www.scopus.com.
[2] http://www.sciencedirect.com.
[3] https://www.ebscohost.com.
[4] https://www.reseaarchgate.net.

Table 2 First search string result

	Scopus	ScienceDirect	EBSCOhost	Total
Number of papers	2097	65	133	2295

Further refinement was performed by (i) limiting the search to journals and conference papers; (ii) selecting computer science and IT related as the subject domain; (iii) selecting ACM, IEEE, SpringerLink, Elsevier as sources; and year of publication to between 2004 and 2018. The year range was selected due to the fact that interest in big data stream analysis actually started in 2004. At this stage, a total of 1989 papers were excluded leaving a total of 315 papers (see Table 3). The result of the search string was exported to PDF.

By going through the title of the papers, 111 seemingly relevant papers were extracted excluding a total number of 213 that were not relevant at this stage (see Table 4).

The abstracts of 111 papers and introduction (for papers that the abstracts were not clear enough) were then read to have a quick overview of the paper and to ascertain whether they are suitable or at variance with the research questions. The citations of the papers were exported to Microsoft Excel for easy analysis. The papers were grouped into three categories; "relevant", "may be relevant" and "irrelevant". The "relevant" papers were marked with black colour, "may be relevant" and "irrelevant" with green and red colours respectively. At the end of this stage, 45 papers were classified as "relevant", 9 papers as "may be relevant" and 11 as "irrelevant". Looking critically at the abstract again, 18 papers were excluded by using the exclusion criteria leaving a total of 47 papers (see Table 5) which were manually reviewed in line with the research questions.

Inclusion criteria

Papers published in journals, peer-reviewed conferences, workshops, technical and symposium from 2004 and 2018 were included. In addition, the most recent papers were selected in case of papers with similar investigations and results.

Table 3 Second search string result

	Scopus	ScienceDirect	EBSCOhost	Total
Number of papers	196	27	92	315

Table 4 Third Search string refinement result

	Scopus	ScienceDirect	EBSCOhost	Total
Number of papers	64	23	24	111

Table 5 Final Selection

	Scopus	ScienceDirect	EBSCOhost	Total
Number of papers	25	10	12	47

Exclusion criteria

Papers that belong to the following categories were excluded from selection as part of the primary study: (i) papers written in source language other than English; (ii) papers with an abstract and or introduction that does not clearly define the contributions of the work; (iii) papers whose abstract do not relate to big data stream analysis.

Result

The findings of the study are now presented with respect to the research questions that guided the execution of the systematic literature review.

Research Question 1: What are the tools and technologies employed for big data stream analysis?

Big data stream platforms provide functionalities and features that enable big data stream applications to develop, operate, deploy, and manage big data streams. Such platforms must be able to pull in streams of data, process the data and stream it back as a single flow. Several tools and technologies have been employed to analyse big data streams. In response to the growing demand for big data streaming analytics, a large number of alternative big data streaming solutions have been developed both by the open source community and enterprise technology vendors. According to [26], there are some factors to consider when selecting big data streaming tools and technologies in order to make effective data management decisions. These are briefly described below.

Shape of the data

Streaming data sources require serialization technologies for capturing, storing and representing such high-velocity data. For instance, some tools and technologies allow projection of different structures across data stores, giving room for flexibility for storage and access of data in different ways. However, the performance of such platforms may not be suitable for high-velocity data.

Data access

There is a need to put into consideration how the data will be accessed by users and applications. For instance, many NoSQL databases require specific application interfaces for data access. Hence there is a need to consider the integration of some other necessary tools for data access.

Availability and consistency requirement

If a distributed system is needed, then CAP theorem states that consistency and availability cannot be both guaranteed in the presence of network partition (i.e. when there is a break in the network). In such a scenario, consistency is often traded off for availability to ensure that requests can always be processed.

Workload profile required

Platform as a service deployment may be appropriate for a spike load profile platform. If platform distribution can be deployed on Infrastructure as a service cloud, then this option may be preferred as users will need to pay only when processing. On-premise

Table 6 Open source tools and technologies for big data stream analysis

Tools and technology	Article
BlockMon	[83]
NoSQL	[4, 84–86]
Spark streaming	[67, 87–91]
Apache storm	[68, 85, 86, 92–97]
Kafka	[85, 91, 95, 96, 98]
Yahoo! S4	[6, 45, 87, 99]
Apache Samza	[46, 67, 100]
Photon	[67, 101]
Apache Aurora	[67, 102]
MavEStream	[103]
EsperTech	[104, 105]
Redis	[106]
C-SPARQL	[107, 108]
SAMOA	[56, 78, 109]
CQELS	[108, 110, 111]
ETALIS	[112]
XSEQ	[73]
Apache Kylin	[113]
Splunk stream	[114]

deployment may be considered for predictable or consistent loads. But if workloads are mixed (i.e. consistent flows or spikes), a combination of cloud and on-premise approach may be considered so as to give room for easy integration of web-based services or software and access to critical functions on the go.

Latency requirement

If a minimal delay or low latency is required, key-value stores may be considered or better still, an in-memory solution which allows the process of large datasets in real-time is required in order to optimize the data loading procedure.

The tools and technologies for big data stream analysis can be broadly categorized into two, which are open source and proprietary solutions. These are listed in Tables 6 and 7.

The selection of big data streaming tools and technologies should be based on the importance of each factor earlier mentioned in this section. Proprietary solutions may not be easily available because of pricing and licensing issues. While open source supports innovation and development at a large scale, careful selection must be made especially when choosing a recent technology still in production due to limited maturity and lack of support from academic researchers or developer communities. In addition, open source solutions may lead to outdating and modification challenges [27]. Moreover, the selection of whether proprietary or open source or combination of both should depend on the problem to address, the understanding of the true costs, and benefits of both open and proprietary solutions.

Table 7 Proprietary tools and technologies for big data stream analysis

Tools and technology	Article
CodeBlue	[115]
Anodot	[116]
Cloudet	[117]
Sentiment brand monitoring	[118]
Numenta	[119]
Elastic streaming processing engine	[120]
Microsoft azure stream analytics	[121]
IBM InfoSphere streams	[8, 122]
Google MillWheel	[123]
Artemis	[124]
WSO2 analytics	[125]
Microsoft StreamInsight	[126]
TIBCO StreamBase	[127]
Striim	[128]
Kyvos insights	[129]
AtScale	[130, 131]
Lambda architecture	[57]

Research Question 2: What methods and techniques are used in analysing big data streams?

Given the real-time nature, velocity and volume of social media streams, the clustering algorithms that are applied on streaming data must be highly scalable and efficient. Also, the dynamic nature of data makes it difficult to know the required or desirable number of clusters in advance. This renders partitioning clustering techniques (such as k-median, k-means and k-medoid) or expectation-maximization (EM) algorithms-based approaches unsuitable for analysing real-time social media data because they require prior knowledge of clusters in advance. In addition, due to concept drift inherent in social media streams, scalable graph partitioning algorithms are not also suitable because of their tendency towards balanced partitioning. Social media streams must be analysed dynamically in order to provide decisions at any given time within a limited space and time window [28–30].

Density-based clustering algorithm (such as DenStream, OpticStream, FlockStream, Exclusive and Complete Clustering) unlike partitioning algorithms does not require apriori number of clusters in advance and can detect outliers [31]. However, the issue with density-based clustering algorithms is that most of them except for few like HDDStream, PreDeCon-Stream and PKS-Stream (which are memory intensive) perform less efficiently in the face of high dimensional data and as a result are not suitable for analyzing social media streams [32].

Threshold-based techniques, hierarchical clustering, and incremental clustering or online clustering are more relevant to social media analysis. Several online threshold-based stream clustering approaches or incremental clustering approaches such as Markov Random Field [33, 34], Online Spherical K-means [35], and Condensed Clusters [36] have been adopted. Incremental approaches are suitable for continuously generated data grouping by setting a maximum similarity threshold between the incoming stream

Table 8 Methods and techniques for big data stream analysis

Methods and techniques	Article
SPADE	[132]
Locally supervised metric learning (LSML)	[133]
KTS	[106]
Multinomial latent dirichlet allocation	[106]
Voltage clustering algorithm	[106]
Locality sensitive hashing (LSH)	[134]
User profile vector update algorithm	[134]
Tag assignment stream clustering (TASC)	[134]
StreamMap	[117]
Density cognition	[117]
QRS detection algorithm	[87]
Forward chaining rule	[110]
Stream	[135]
CluStream	[136, 137]
HPClustering	[138]
DenStream	[139]
D-Stream	[140]
ACluStream	[141]
DCStream	[142]
P-Stream	[143]
ADStream	[144]
Continuous query processing (CQR)	[145]
FPSPAN-growth	[146]
Outlier method for cloud computing algorithm (OMCA)	[147]
Multi-query optimization strategy (MQOS)	[148]
Parallel K-means clustering	[72]
Visibly push down automata (VPA)	[73]
Incremental MI outlier detection algorithm (Inc I-MLOF)	[149]
Adaptive windowing based online ensemble (AWOE)	[74]
Dynamic prime-number based security verification	[84]
K-anonymity, l-diversity, t-closeness	[90]
Singular spectrum matrix completion (SS-MC)	[76]
Temporal fuzzy concept analysis	[96]
ECM-sketch	[77]
Nearest neighbour	[91]
Markov chains	[91]
Block-QuickSort-AdjacentJobMatch	[86]
Block-QuickSort-OverlapReplicate	[86]
Fuzzy-CSar-AFP	[150]
Weighted online sequential extreme learning machine with kernels (WOS-ELMK)	[22]
Concept-adapting very fast decision tree (CVFDT)	[151]

and the existing clusters. Much work has been done in improving the efficiency of online clustering algorithms, however, little research efforts have been directed to threshold and fragmentation issues. Incremental algorithm threshold setting should employ adaptive approach instead of relying on static values [37, 38]. Some of the methods and techniques that have been employed in analysing big data streams are outlined in Table 8.

Many researchers have looked at the aspect of the real-time analysis of big data streams but not much attention has been directed towards social media stream preprocessing. For instance, the social media stream is characterized by incomplete, noisy, slang, abbreviated words. Also, contextual meaning of social media post is essential for improved event detection, sentiment analysis or any other social media analytics algorithms in terms of quality and accuracy [36, 39]. There is the need to give more attention to the preprocessing stage of social media stream analysis in the face of incomplete, noisy, slang, and abbreviated words that are pertinent to social media streams. These challenges create opportunities application of new semantic technology approaches, which are more suited to social media streams [40, 41].

Research Question 3: What do big data streaming tools and technologies have in common and their differences in terms of concept, purpose, and capabilities?

The features of various tools and technologies for big data stream were compared in order to answer this question. An overview analysis based on 10 dimensions, which are database support, execution model, workload, fault-tolerance, latency, throughput, reliability, operating system, implementation languages and application domain or areas is presented in Table 9.

For organisations with existing applications that have support for SQL, MySQL, SQL Server, Oracle Database, for instance, may consider choosing big data streaming tools and technologies that have support for their existing databases. There are few big data streaming tools and technology that support virtually any data format. An example of such is Infochimps Cloud.

The major big data streaming tools and technologies considered are all suitable for streaming execution model, however out of 19 big data tools and technology compared and contrasted in this section, only 10.5% is suitable for streaming, batch, and iterative processing while 47.4% can handle jobs requiring both batch and streaming processing. It is safer for a job to be executed on a single platform which can accommodate all the dependencies required in order to avoid interoperability constraints than combining two or more platforms or frameworks. The best fit with respect to the choice of big data streaming tools and technologies will depend on the state of data to process, infrastructure preference, business use case, and kind of results interested in.

Virtually all the big data streaming tools and technologies are memory intensive. This implies that the main performance bottleneck at higher load conditions will be due to lack of memory [42]. However, research has shown that the benefit of high intensive memory applications outweighs the performance loss due to long memory latency [43].

From all the big data streaming tools and technologies reviewed, only IBMInfoSphere and TIBCO StreamBase support all of the three "at-most-once" "at-least-once" and "exactly-once" message delivery mechanisms while others support one or two of the three delivery mechanisms. "At-most-once" is the cheapest with least implementation overhead and highest performance because it can be done in a fire-and-forget fashion without keeping the state in the transport mechanism or at the sending end. "At-least-once" delivery requires multiple attempts in order to counter transport losses which means keeping the state at the sending end and having an acknowledgement mechanism at the receiving end. "Exactly-once" is the most expensive and has consequently worst

Table 9 Comparison of big data streaming tools and technologies

Tools and technology	Database support	Execution model	Workload	Fault tolerance	Latency	Throughput	Reliability	Operating system	Implementation/ supported languages	Application
BlockMon	Cassandra, MongoDB, XML	Streaming	Multi-slice memory allocation and batch allocations	Checkpoint, rollback	Very low	High	At least once	Linux	C ++11, Python	Anomaly detection, network optimization, multimedia content delivery, financial market analysis, web analytics
Spark Streaming	Kafka, HBase, Hive Flume, HDF/S3, Kinesis, TCP sockets, Twitter, SQL	Batch, Iterative, Streaming	CPU/memory intensive	RDD based Check-pointing, parallel recovery, replication	Low	High	Exactly once	Windows, macOS, Linux	Scala, Python, Java, R	Event detection, streaming machine learning, fog computing, interactive analysis, multimedia analysis, cluster analysis, filtering, re-processing, cache invalidation
Apache Storm	Spout, HBase, Hive, SQL, Cassandra, Memcached	Streaming	CPU/memory intensive	Replication, checkpoint, data recovery, Upstream backup, record-level acknowledgement, stateless management	Very low	Low	At least once	Windows, macOS, Linux	Clojure, Java, Scala, Clojure, non-JVM languages	Internet of things, streaming machine learning, multimedia analysis
Yahoo! S4	MySQL, NoSQL, Rich Data Format	Streaming	CPU/memory intensive	Replication, checkpoint, data recovery	Low	Low	Exactly once	Linux	Java, Python, C++, Perl	Online analytics, monitoring, fraud detection, financial data processing, web personalization and session modelling

Table 9 (continued)

Tools and technology	Database support	Execution model	Workload	Fault tolerance	Latency	Throughput	Reliability	Operating system	Implementation/ supported languages	Application
Apache Samza	Kafka, HDFS, Kinesis, Stream consumer, Key-value stores	Streaming, batch processing	Memory intensive	Checkpoint	Very low	High	At least once	Linux, Windows	Java, Scala, JVM languages	Filtering, re-processing, cache invalidation
Apache Flink	Kafka, Flume, HDF/S3, Kinesis, TCP sockets, Twitter, Cassandra, Redis, MongoDB, HBase, SQL	Streaming, batch, iterative, interactive	Memory intensive	Stream replay and marker-checkpoint	Very low	High	Exactly once	Linux, MacOS, Windows	Java, Scala, Python	Optimization of e-commerce search result, network/sensor monitoring and error detection, ETL for business intelligence infrastructure, machine learning
Apache Aurora	H2, Java maps, MyBatis, MySQL, PostgreSQL	Streaming	Memory and disk space	Periodic recovery checkpoint and rollback	Low	High	At least once	Linux	Python	Monitoring applications such as financial analysis and military applications
Redis	Key-value stores, rabitmq, MongoDB	Streaming	In-memory but persistent on-disk database	Replica migration, Sentinel	Low	High	At least once	Ubuntu, Linux, OSX	C, C#, Java, PHP, Python	Web analysis, cache, message queues
C-SPARQL	RDF, SQLJ, NoSQL, HDF	Batch, streaming	Low memory usage	Adaptation	Very low	High	Cumulative	Windows, Linux, MacOS, Android	Java, Apache Jena libraries	Real-time reasoning over sensor data, social semantic data, urban computing
SAMOA	HBase, Hive, Cassandra	Streaming	Low memory usage	Upstream backup	Low	High	Exactly once	Linux	Java	Classification, clustering, spam detection, regression, frequent pattern mining

Table 9 (continued)

Tools and technology	Database support	Execution model	Workload	Fault tolerance	Latency	Throughput	Reliability	Operating system	Implementation/ supported languages	Application
CQELS	RDF, SQLJ, NoSQL, HDF	Batch, streaming	In-memory	Adaptation	Low	High	Cumulative	Windows, Linux, MacOS, Android	Java	Real-time reasoning over sensor data, social semantic data, urban computing
ETALIS	RDF	Streaming	Binarization	Adaptation	Low	Low	Cumulative	Windows, Linux, MacOS, Android	Prolog, Java, C, SPARQL, C#, ETALIS Language for Events (ELE)	Event detection, reasoning over streaming events
XSEQ	XML	Batch, streaming	In-memory with buffering	checkpoint	Low	High	At least once	Windows, Linux	Java, Apache Xerces	Biological data, social networks, user behaviour, financial data analysis, filtering
IBM InfoSphere streams	Pig, Hive, Jaql, HBase Flume, Lucene, Avro, ZooKeeper, Oozie, Oracle Database, DB2, Netezza, MySQL, Aster, Informix.	Streaming	Capture database workloads and replay them in a test database environment	Automatic recovery	Low	High	Exactly once, At least once, At most once	Linux, CentOS	C++ Java SPL	Space weather prediction, physiological data streams analysis, traffic management, real-time predictions, event detection, visualisation
Google Mill-Wheel	BigTable, Spanner	Streaming	In-memory and bloom filtering	Uncoordinated periodic, checkpoint, upstream backup	Low	High	Exactly once	Linux	Virtually any programming language	Anomaly detection, health monitoring, image processing, network switch management

Table 9 (continued)

Tools and technology	Database support	Execution model	Workload	Fault tolerance	Latency	Throughput	Reliability	Operating system	Implementation/ supported languages	Application
Infochimps cloud	SQL, NoSQL, Hive, Pig Wukong, Hadoop, RDBMS, Virtually any data format	Batch, streaming	In-memory	Upstream backup	Low	High	Exactly once	Linux	Java	Disaster discovery, text analysis, complex event processing, visualisation
Microsoft StreamInsight	SQL Server	Streaming	In-memory	Replication, checkpoint, data recovery	Very low	High	Exactly once	Windows	.NET, C#, LINQ, Rx	Manufacturing process monitoring and control, financial data analysis, operation analytics, web analytics, event pattern detection
TIBCO StreamBase	Oracle database, SQL Server, Impala	Batch, Streaming	In-memory	Synchronization, replication, rollback	Very low	High	At least once/ at most once/ exactly once	Windows, MacOS, Linux	R, Java	Mission critical analysis, IoT analysis, click-stream analysis, predictive analytics, workflow optimization, risk avoidance
Lambda Architecture	RDBMS, Cassandra, Kafka, Data Warehouses, Kinesis Data Stream, HDFS, HBase	Batch, Streaming	In-memory/disk database	Replication, checkpoint	Low	Low	Exactly once	Ubuntu, Windows, Linux	Java, C#, Python, Pig Latin	IoT analysis, tracking real-time updates, financial risk management, click-stream analysis

performance because, in addition to "at-least-once" delivery mechanism, it requires the state to be kept at the receiving end in order to filter duplicate deliveries. In other words, "at-most-once" delivery mechanism implies that the message may be lost while "at-least-once" delivery ensures that messages are not lost and "exactly-once" implies that message can neither be lost nor duplicated. "Exactly-once" is suitable for many critical systems where duplicate messages are unacceptable.

Research Question 4: What are the limitations and strengths of big data streaming tools and technologies?

Observations from the literature reveal that specific big data streaming technology may not provide the full set of features that are required. It is rare to find specific big data technology that combines key features such as scalability, integration, fault-tolerance, timeliness, consistency, heterogeneity and incompleteness management, and load balancing. For instance, Spark streaming [16] and Sonora [44] are excellent and efficient for checkpointing but the operator space available to user codes are limited. S4 does not guarantee 100% fault-tolerant persistent state [45]. Storm does not guarantee the ordering of messages due to its "at-least-once" mechanism for record delivery [46, 47]. Strict transaction ordering is required by Trident to operate [48]. While streaming SQL provide simple and succinct solutions to many streaming problems, the complex application logic (such as matrix multiplication) and intuitive state abstractions are expressed with the operational flow of an imperative language rather than a declarative language such as SQL [49–51].

Moreover, BlockMon uses batches and cache locality optimization techniques for memory allocation efficiency and data speed up access. However, deadlock may occur if data streams are enqueued with a higher rate than that of the block consumption [52]. Apache Samza solves batch latency processing problems but requires an added layer for flow control [53]. Flink is suitable for heavy stream processing and batch-oriented tasks although it has scaling limitations [46]. Redis' in-memory data store makes it extremely fast although this implies that available memory size determines the size of the Redis data store [54]. While C-SPARQL and CQELS are excellent for combining static and streaming data, they are not suitable when scalability is required [55]. SAMOA is suitable for machine learning paradigm as it focuses on speed/real-time analytics, scales horizontally and is loosely coupled with its underlying distributed computation platform [56]. With Lambda architecture, a real-time layer can complement the batch processing one thereby reducing maintenance overhead and risk for errors as a result of duplicate code bases. In addition, Lambda architecture handles reprocessing, which is one of the key challenges in stream processing. Two main problems with Lambda architecture are code maintenance in two complex distributed systems that need to produce the same result and high operational complexity [57, 58].

Summarily, there exists various tools and technologies for implementing big data streams and there seems to be no big data streaming tool and technology that offers all the key features required for now. While each tool and technology may have its strengths and weaknesses, the choice depends on the objective of the research and data availability. A decision in favour of the wrong technology may result in increased overhead cost and time. The decision should take into consideration empirical analysis along with

system requirements. In addition, research efforts should also be directed to how to improve on existing big data streaming tools and technologies to provide key features such as scalability, integration, fault-tolerance, timeliness, consistency, heterogeneity and incompleteness management, and load balancing.

Research Question 5: What are the evaluation techniques or benchmarks that are used for evaluating big data streaming tools and technologies?

The diversity of big data poses a challenge when it comes to developing big data benchmarks that will be suitable for all workload cases. One cannot stick to one big data benchmark because it has been observed that using only one benchmark on different data sets do not give the same result. This implies that benchmark testing should be application specific. Subsequently, in evaluating big data system, the identification of workload for an application domain is a prerequisite [59]. Most of the existing big data benchmarks are designed to evaluate a specific type of systems or architectures. For instance, HiBench [60] is suitable for benchmarking Hadoop, Spark and streaming workloads, GridMix [61] and PigMix [62] are for MapReduce Hadoop systems. BigBench [63, 64] is suitable for benchmarking Teradata Aster DBMS, MapReduce systems, Redshift database, Hive, Spark and Impala. Presently, BigDataBench [65, 66] seems to be the only big data benchmark that can evaluate a hybrid of different big data systems.

So far, many researchers have evaluated their work by making use of synthetic and real-life data. Standard benchmark dataset for big data streaming analytics has not been widely adopted. However, few of the researchers that used standardized benchmarking are briefly discussed below. The work of [67] was tested with two benchmarks; Word Count and Grep. The result showed that the proposed algorithm can effectively handle unstable input and the delay of the total event can be limited to an expected range.

The tool developed by [68] was tested on both car dataset and Wikinews[5] dataset in comparison with sequential processing. It was discovered that their tool (pipeline implementation) performed better and faster.

Krawczyk and Wozniak used several benchmark datasets which include Breast-Wisconsin, Pima, Yeast3, Voting records, CYP2C19 isoform, RBF for estimating weights for the new incoming data stream with their proposed method against other standard methods. They also analysed time and memory requirements. Experimental investigation result proved that the proposed method can achieve better [69].

A benchmark evaluation using an English movie review dataset collected from Rotten Tomatoes website (a de facto benchmark for analysing sentiment applications) was conducted by [70], the result showed that sentiment analysis engine (SAE) proposed by the authors outperformed the bag of words approach.

Authors' suite of ideas in [71] outperformed state-of-the-art searching technique called EBSM. The work of [72] used various datasets such as KDD-Cup 99, Forest Cover type, Household power consumption, etc. They compared their algorithm—parallel K-means clustering with k-means and k-means++, the result showed that their algorithm performed better in terms of speed.

[5] http://en.wikinews.org.

Mozafari et al. in [73] benchmarked their system, XSeq against other general-purpose XML engines. The system outperformed other complex event processing engines by two orders of magnitude improvement.

Authors in [74] evaluated their work in terms of time, accuracy and memory using Forest cover type, Poker hand, and electricity datasets. They compared their method, adaptive windowing based online ensemble (AWOE) with other standard methods such as accuracy updated ensemble (AUE), online accuracy updated ensemble (OAUE), accuracy weighted ensemble (AWE), dynamic weighted majority (DWM) and Lev Bagging (Lev). Their proposed approach outperformed other methods in three perspectives which include suitability in terms of different type of drifts, better resolved appropriate size of block, and efficiency.

The evaluation performed by [75] using FACup and Super Tuesday datasets showed that their method, which is a hybrid of topic extraction methods (i.e. a combination of feature pivot and document pivot) has high efficiency and accuracy with respect to recall and precision.

Evaluating the performance of low-rank reconstruction and prediction scheme, specifically, singular spectrum matrix completion (SS-MC) proposed by [76], SensorScope Grand St-Bernard dataset[6] and Intel Berkeley Research Lab dataset[7] were used. The authors compared their proposed method with three state-of-the-art methods; KNN-imputation, RegEM and ADMM version of MC and discovered that their method outperformed the other methods in terms of pure reconstruction as well as in the demanding case of simultaneous recovery and prediction.

The authors in [77] evaluated their work using World Cup 1998 and CAIDA Anonymized Internet Traces 2011 datasets. When their method, ECM-Sketch (a sketch synopsis that allows effective summarization of streaming data over both time-based and count-based sliding windows) was compared with three state-of-the-art algorithms (Sketch variants); ECM-RW, ECM-DW, and ECM-EH, variants using randomized waves, deterministic waves and exponential histograms respectively, their method reduce memory and computational requirements by at least one order of magnitude with a very small loss in accuracy.

The work of [78] centred on benchmarking real-time vehicle data streaming models for a smart city using a simulator that emulates the data produced by a given amount of simultaneous drivers. Experiment with the simulator shows that streaming processing engine such as Apache Kafka could serve as a replacement to custom-made streaming servers to achieve low latency and higher scalability together with cost reduction.

A benchmark among Kyvos Insight, Impala and Spark conducted by [79] shows that Kyvos Insight performed analytical queries with much lower latencies when there is a large number of concurrent users due to pre-aggregation and incremental code building [80].

Authors in [81] proposed that in addition to execution time and resource utilization, microarchitecture-level and energy consumption are key to fully understanding the behaviour of big data frameworks.

[6] http://lcav.epfl.ch.page-86035-en.html.

[7] http://db.csail.mit.edu/labdata/labdata.html.

In addition, to strengthen the confidence of big data research evaluation or result, application of empirical methods (i.e. tested or evaluated concept or technology for evidence-based result) should be highly encouraged. The current status of empirical research in big data stream analysis is still at an infant stage. The maturity of a research field is directly proportional to the number of publications with empirical result [20, 21]. According to [21] that conducted a systematic literature mapping to verify the current status of empirical research in big data, it was found out that only 151 out of 1778 studies contained empirical result. As a result, more research efforts should be directed to empirical research in order to raise the level of confidence of big data research outputs than it is at present.

Moreover, only a few big data benchmarks are suitable for different workloads at present. Research efforts should be geared towards advancing benchmarks that are suitable for evaluating different big data systems. This would go a long way to reduce cost and interoperability issue.

Discussion

From the analysis, it was observed that there has been a wave of interest in big data stream analysis since 2013. The number of papers produced in 2012 was doubled in 2013. In the same vein, more than double of the papers in 2013 were produced in 2014. There was a relative surge in 2017 having a total of 98 paper while the year 2018 received 156 papers (see Tables 9, 10 and Fig. 2). The percentage of papers analyzed from journals was 50%; that of conferences was 41% while that of workshop/technical/symposium was 9% as depicted in Fig. 3. Figure 4 presented the frequency of research efforts from different geographical locations with researchers from China taking the lead.

The selection of big data streaming tools and technologies should be based on the importance of each of the factors such as the shape of the data, data access, availability and consistent requirements, workload profile required, and latency requirement. Careful selection with respect to open source technology must be made especially when choosing a recent technology still in production. Moreover, the problem to address, the understanding of the true costs, and benefits of both open and proprietary solutions are also vital when making a selection.

A lot of research efforts have been directed to big data stream analysis but social media stream preprocessing is still an open issue. Due to inherent characteristics of social media stream which include incomplete, noisy, slang, abbreviated words, social media streams present a challenge to big data streams analytics algorithms. There is the need to give more attention to the preprocessing stage of social media stream analysis in the face of incomplete, noisy, slang, and abbreviated words that are pertinent to social media streams in order to improve big data streams analytics result.

Out of 19 big data streaming tools and technologies compared, 100% support streaming, 47.4% can do both batch and streaming processing while only 10.5% support streaming, batch and iterative processing. Depending on the state of the data to be processed, infrastructure preference, business use case, and kind of results that is of interest, choosing a single big data streaming technology platform that supports all the system requirements minimizes the effect of interoperability constraints.

From all the big data streaming tools and technologies reviewed, only IBMInfoSphere and TIBCO StreamBase support all of the three "at-most-once", "at-least-once", and "exactly-once" message delivery mechanisms while others support one or two of the three delivery mechanisms. Having all the three delivery mechanisms give room for flexibility.

It is rare to find a specific big data technology that combines key features such as scalability, integration, fault-tolerance, timeliness, consistency, heterogeneity and incompleteness management, and load balancing. There seems to be no big data streaming tool and technology that offers all the key features required for now. This calls for more research efforts that are directed to building more robust big data streaming tools and technologies.

Few big data benchmarks are suitable for a hybrid of big data systems at present and standard benchmark datasets for big data streaming analytics have not been widely adopted. Hence, research efforts should be geared towards advancing benchmarks that are suitable for evaluating different big data systems.

Limitation of the review

While authors explored Scopus, ScienceDirect and EBSCO databases which index high impact journals and conference papers from IEEE, ACM, SpringerLink, and Elsevier to identify all possible relevant articles, it is possible that some other relevant articles from other databases such as Web of Science could have been missed.

The analysis and synthesis are based on interpretation of selected articles by the research team. The authors attempted to avoid this by cross-checking papers to deal with bias though that cannot completely rule out the possibility of errors. In addition, the authors implemented the inclusion and exclusion criteria in the selection of articles and only relevant articles written in the English Language were selected. Building on the underpinning of the findings of the research, while a lot of research has been done with respect to tools and technologies as well as methods and techniques employed in big data streaming analytics, method of evaluation or benchmarks of the technologies of various workloads for big data streaming analytics have not received much attention. As it could be gathered from the literature reviewed that most of the researchers evaluated their work using either synthetic or real-life datasets.

Conclusion and further work

As a result of challenges and opportunities presented by the Information Technology revolution, big data streaming analytics has emerged as the new frontier of competition and innovation. Organisations who seize the opportunity of big data streaming analytics are provided with insights for robust decision making in real-time thereby making them to have an edge over their competitors.

In this paper, the authors have tried to present a holistic view of big data streaming analytics by conducting a comprehensive literature review to understand and identify the tools and technologies, methods and techniques, benchmarks or methods of evaluation employed, and key issues in big data stream analysis to showcase the signpost of future research directions.

Table 10 Distribution of papers over the studied years

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	Total
Paper	2	1	2	3	5	2	5	4	5	10	22	28	38	98	156	381

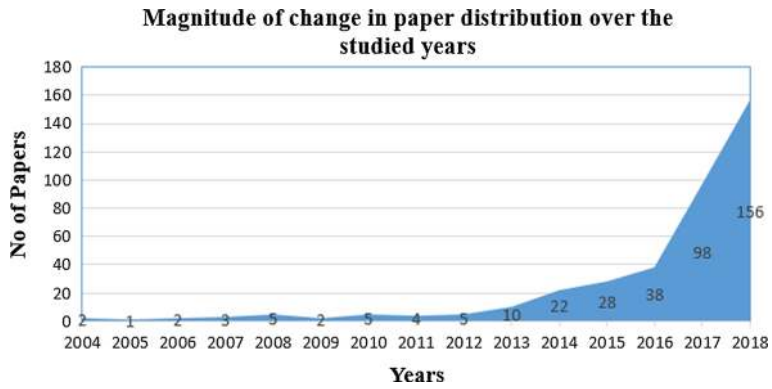

Fig. 2 Magnitude of change in paper distribution. The figure shows the magnitude of change in paper distribution over the studied years (i.e. 2004 to 2018)

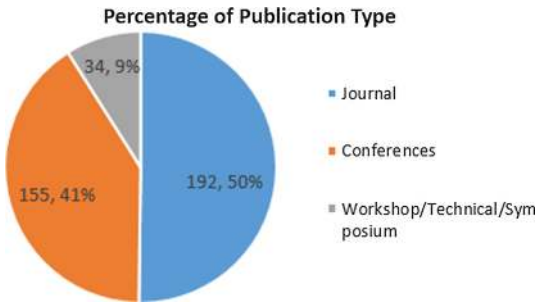

Fig. 3 Percentage of publication type. The figure shows percentage of 381 papers from journals (50%), conferences (41%), and workshop/technical/symposium (9%)

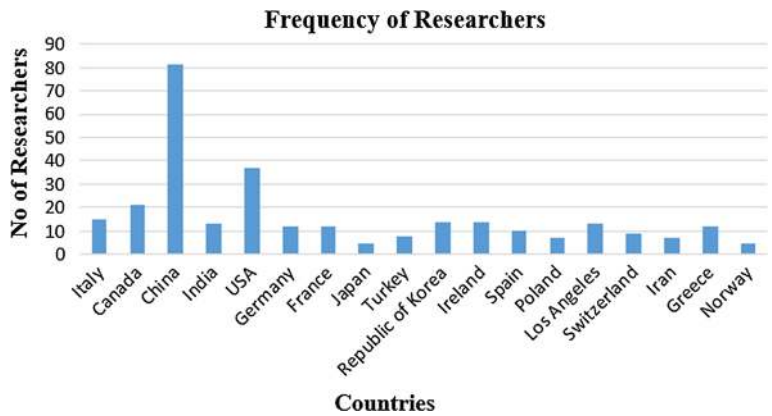

Fig. 4 Frequency of researchers across different. The figure presented the frequency of research geographical locations research efforts from different geographical locations

Although a lot of research efforts have been directed towards big data at rest (i.e. big data batch processing), there has been increased interest in analysing big data in motion (i.e. big data stream processing). With respect to issues identified in this paper, big data streaming analytics can be considered as an emerging phenomenon although some countries and industries have seized the opportunities by making it a

pertinent research area. Some of the key issues such as scalability, integration, fault-tolerance, timeliness, consistency, heterogeneity and incompleteness, load balancing, high throughput, and privacy that require further research attention were identified. While researchers have invested a lot of efforts to mitigate these issues, scalability, privacy and load balancing remain a concern. In addition, researchers also need to give more focus to the empirical analysis of big data streaming tools and technologies in order to be able to provide concrete reasons and support for choosing a tool/technology based on empirical evidence.

Presently, BigDataBench seems to be the only big data benchmark that can evaluate a hybrid of different big data systems. Standard benchmark for a hybrid of big data systems has not been widely adopted. It is rare to find a specific big data technology that combines key features such as scalability, integration, fault-tolerance, timeliness, consistency, heterogeneity and incompleteness management, and load balancing.

There is the need to give more attention to the preprocessing stage of social media stream analysis in the face of incomplete, noisy, slang, and abbreviated words that are pertinent to social media streams. Many researchers have looked at the aspect of the real-time analysis of big data streams but not much attention has been directed towards social media stream preprocessing.

In addition, research efforts should be geared towards developing scalable frameworks and algorithms that will accommodate data stream computing mode, effective resource allocation strategy and parallelization issues to cope with the ever-growing size and complexity of data. As regards load balancing, a distributing environment that automatically streams partial data streams to a global centre when local resources become insufficient is required. The demand for big data stream analysis is that data must be analysed as soon as they arrive makes privacy issue a big concern. The main challenge here is proposing techniques for protecting a big data stream dataset before its analysis in such a way that the real-time analysis is still maintained. As a result, research efforts should be directed to the identified areas in order to have robust solutions for big data streaming analytics.

Abbreviations

AUE: accuracy updated ensemble; AWE: accuracy weighted ensemble; AWOE: adaptive windowing based online ensemble; CEP: complex event processing; CQR: continuous query processing; DWM: dynamic weighted majority; EM: expectation–maximization; GOAL: GOd's ALgorithm; IDC: International Data Cooperation; Inc I-MLOF: Incremental MI outlier detection algorithm; LSH: locality sensitive hashing; LSML: locally supervised metric learning; MQOS: multi-query optimization strategy; OAUE: online accuracy updated ensemble; OMCA: outlier method for cloud computing algorithm; SAE: sentiment analysis engine; SAMOA: scalable advanced massive online analysis; SS-MC: singular spectrum matrix completion; VPA: visibly push down automata; WOS-ELMK: weighted online sequential extreme learning machine with kernels.

Acknowledgements

The research was supported by Covenant University Centre for Research, Innovation, and Discovery (CUCRID); Landmark University, Omu-Aran, Osun State, Nigeria; The World Academy of Sciences for Research and Advanced Training Fellowship, FR Number: 3240301383; and the Cape Peninsula University of Technology, South Africa.

Authors' contributions

TK gathered all the papers from various databases that were used for the manuscript and was a major contributor in writing the manuscript. OD ensured that the guideline for systematic review literature was followed, and provided direction for the literature-based research. AA contributed to the validation of selected primary studies. All authors contributed to the selection of papers for the systematic review. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials
All data (papers) analysed are included in Scopus, ScienceDirect, and EBSCOhost.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Department of Computer and Information Sciences, Covenant University, Ota, Nigeria. [2] Department of Computer Science, Federal University Lokoja, Lokoja, Kogi, Nigeria. [3] Department of Information Technology, Cape Peninsula University of Technology, Cape Town, South Africa. [4] Department of Computer Science, Landmark University, Omu-Aran, Kwara, Nigeria.

Received: 28 March 2019 Accepted: 28 May 2019
Published online: 06 June 2019

References

1. Mavragani A, Ochoa G, Tsagarakis KP. Assessing the methods, tools, and statistical procedures in Google trends research: systematic review. J Med Internet Res. 2018;20(11):e270.
2. Sun D, Zhang G, Zheng W, Li K. Key technologies for big data stream computing. In: Li K, Jiang H, Yang LT, Guzzocrea A, editors. Big data algorithms, analytics and applications. New York: Chapman and Hall/CRC; 2015. p. 193–214. ISBN 978-1-4822-4055-9.
3. Qian ZP, He Y, Su CZ et al. TimeStream: Reliable stream computation in the cloud. In: Proc. 8th ACM European conference in computer system, EuroSys 2013. Prague: ACM Press; 2013. p. 1–4.
4. Liu R, Li Q, Li F, Mei L, Lee, J. Big data architecture for IT incident management. In: Proceedings of IEEE international conference on service operations and logistics, and informatics (SOLI), Qingdao, China. 2014. p. 424–9.
5. Sakr S. An introduction to Infosphere streams: A platform for analysing big data in motion. IBM. 2013. https://www.ibm.com/developerworks/library/bd-streamsintro/index.html. Accessed 7 Oct 2018.
6. Xhafa F, Naranjo V, Caballé S. Processing and analytics of big data stream with Yahoo!S4. In: 2015 IEEE 29th international conference on advanced information networking and applications, Gwangiu, South Korea, 24–27 March 2015. 2015. https://doi.org/10.1109/aina.2015.194.
7. Marz N. Storm: distributed and fault-tolerant real-time computation. In: Paper presented at Strata conference on making data work, Santa Clara, California, 28 Feb–1 March 2012. 2012. https://cdn.oreillystatic.com/en/assets/1/event/75/Storm_%20distributed%20and%20fault-tolerant%20realtime%20computation%20Presentation.pdf. Accessed 25 Jan 2018.
8. Ballard C, Farrell DM, Lee M, Stone PD, Thibault S, Tucker S. IBM InfoSphere Streams: harnessing data in motion. IBM Redbooks. 2010.
9. Joseph S, Jasmin EA, Chandran S. Stream computing: opportunities and challenges in smart grid. Procedia Technol. 2015;21:49–53.
10. IBM Research (no date) Stream computing platforms, applications and analytics. IBM. http://researcher.watson.ibm.com/researcher/view_grp.php?id=2531 Accessed 5 Mar 2019.
11. Gantz J, Reinsel D. The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the Far East. New York: IDC iView: IDC Analyse future; 2012.
12. Cortes R, Bonnaire X, Marin O, Sens P. Stream processing of healthcare sensor data: studying user traces to identify challenges from a big data perspective. The 4th international workshop on body area sensor networks (BASNet-2015). Procedia Comput Sci. 2015;52:1004–9.
13. Chung D, Shi H. Big data analytics: a literature review. J Manag Anal. 2015;2(3):175–201.
14. Lu J, Li D. Bias correction in a small sample from big data. IEEE Trans Knowl Data Eng. 2013;25(11):2658–63.
15. Garzo A, Benczur AA, Sidlo CI, Tahara D, Ywatt EF. Real-time streaming mobility analytics. In: Proc. 2013 IEEE international conference on big data, big data, Santa Clara, CA, United States, IEEE Press. 2013. p 697–702.
16. Zaharia M, Das T, Li H, Hunter T, Shenker S, Stoica I. Discretized streams: fault-tolerant streaming computation at scale. In: Proc. the 24th ACM symposium on operating system principles, SOSP 2013, Farmington, PA, United States. New York: ACM Press; 2013. p. 423–38.
17. Fan J, Liu H. Statistical analysis of big data on pharmacogenomics. Adv Drug Deliv Rev. 2013;65(7):987–1000.
18. Bifet A, Holmes G, Kirkby R, Pfahringer B. Moa: massive online analysis. J Mach Learn Res. 2010;11:1601–4.
19. Akter S, Fosso WS. Big data analytics in e-commerce: a systematic review and agenda for future research. Electr Markets. 2016;26:173–94.
20. Sivarajah U, Kamal MM, Irani Z, Weerakkody V. Critical analysis of big data challenges and analytical methods. J Bus Res. 2016;70:263–86.
21. Wienhofen LW, Mathisen BM, Roman D. Empirical big data research: a systematic literature mapping. CoRR, abs/1509.03045. 2015.
22. Habeeb RAA, Nasaruddin F, Gani A, Hashem IAT, Ahmed E, Imran M. Real-time big data processing for anomaly detection: a survey. Int J Inform Manage. 2018;45:289–307. https://doi.org/10.1016/j.ijinfomgt.2018.08.006.
23. Mehta N, Pandit A. Concurrence of big data analytics in healthcare: a systematic review. Int J Med Inform. 2018;114:57–65.
24. Kitchenham BA, Charters S. Guidelines for performing systematic literature review in software engineering. Technical report 2(3), EBSE-2007-01, Keele University and University of Durham. 2007.
25. Host M, Orucevic-Alagic A. A systematic review of research on open source software in commercial software product development. 2013. http://www.bcs.org/upload/pdf/ewic_ea10_session5paper2.pdf. Accessed 2 Mar 2018.

26. Millman N. Analytics for business. Computerworld. 2014. https://www.computerworld.com/article/2475840/big-data/8-considerations-when-selecting-big-data-technology.html. Accessed 7 Oct 2018.
27. Oussous A, Benjelloun F, Lachen AA, Belfkih S. Big data technologies: a survey. J King Saud Univ Comput Inform Sci. 2018;30:431–48.
28. Becker H, Naaman M, Gravano L. Learning similarity metrics for event identification in social media. In: Proceedings of the third ACM international conference on web search and data mining (WSDM'10), ACM New York, NY, USA, 4–6 Feb 2010. 2010. p. 291–300.
29. Aggarwal CC, Zhai C. A survey of text clustering algorithms. In: Aggarwal CC, Zhai C, editors. Mining text data. New York: Springer; 2012. p. 77–128.
30. Panagiotou N, Katakis I, Gunopulos D. Detecting events in online social networks: Definitions, trends and challenges. In: Michaelis S, et al., editors. Solving large scale learning tasks: challenges and algorithms. Lecture Notes in Computer Science, vol. 9850. Cham: Springer; 2016. p. 42–84. https://doi.org/10.1007/978-3-319-41706-6_2.
31. Deepa MS, Sujatha N. Comparative study of various clustering techniques and its characteristics. Int J Adv Netw Appl. 2014;5(6):2104–16.
32. Reddy KSS, Bindu CS. A review of density-based clustering algorithms for big data analysis. In: International conference on I-SMAC (IoT in Social, Mobile, Analytic, and Cloud), Palladam, India 10–11 February 2017, IEEE. 2017. https://doi.org/10.1109/i-smac.2017.8058322.
33. Pelkowitz L. A continuous relaxation labelling algorithm for Markov random fields. IEEE Trans Syst Man Cybern. 1990;20:709–15.
34. Li SZ. Markov random field modelling in image analysis. New York: Springer; 2001.
35. Zhong S. Efficient streaming text clustering. Neural Netw. 2005;18:5–6.
36. Aggarwal CC, Yu PS. A framework for clustering massive text and categorical data streams. In: Proceedings of the sixth SIAM international conference on data mining, Bethesda, MD, USA, 20–22 Apr 2016. 2006. https://doi.org/10.1137/1.97816119727644.
37. Li H, Jiang X, Xiong L, Liu J. Differentially private histogram publication for dynamic datasets: an adaptive sampling approach. Proc ACM Int Conf Knowl Manag. 2015. p. 1001–10. https://doi.org/10.1145/2806416.2806441.
38. Deng JD. Outline detection energy data streams using incremental and kernel PCA algorithms. 2016 IEEE 16th international conference on data mining workshops. 2016. p. 390–7. https://doi.org/10.1109/icdmw.2016.158.
39. Limsopatham N, Collier N. Adapting phrase-based machine translation to normalise medical terms in social media messages. In: Proceedings of the 2015 conference on empirical methods in natural language processing, EMNLP 2015, Lisbon. 2015. p.ρ 1675–80.
40. Kaushik R, Apoorva CS, Mallya D, Chaitanya JNVK, Kamath SS. Sociopedia: an interactive system for event detection and trend analysis for Twitter data. In: Nagar A, Mohapatra D, Chaki N (eds) Smart innovation, systems and technologies, proceedings of 3rd international conference on advanced computing, networking and informatics. New Delhi: Springer; 2016.
41. Carter S, Weerkamp W, Tsagkias E. Microblog language identification: overcoming the limitations of short, unedited and idiomatic text. Lang Resour Eval J. 2013;47(1):195–215.
42. Pooja P, Pandey A. Impact of memory intensive applications on performance of cloud virtual machine. In: Proceedings of 2014 recent advances in engineering and computational sciences (RAECS), UIET Panjab University Chandigarh, 6–8 March 2014. 2014. p. 1–6. https://doi.org/10.1109/raecs.2014.6799629.
43. Chang M, Choi IS, Niu D, Zheng H. Performance impact of emerging memory technologies on big data applications: a latency-programmable system emulation approach. In: Proceedings of 2018 on great lake symposium on VLSI (GLSVLSI'18), Chicago, IL, USA, ACM New York, NY, USA, 23–25 May 2018. 2018. p. 439–42. https://doi.org/10.1145/3194554.3194633.
44. Yang W, Da Silva A, Picard ML. Computing data quality indicators on big data streams using a CEP. In: International workshop on computational intelligence for multimedia understanding IWCIM, Prague, Czech Republic, 29–30 October 2015. 2015.
45. Neumeyer L, Robbins B, Nair A, Kesari A. S4: Distribute stream computing platform. In: Proceedings of the 2010 IEEE international conference on data mining workshops. 2010. p. 170–7. https://doi.org/10.1109/icdmw.2010.172.
46. Inoubli W, Aridhi S, Mezni H, Maddouri M, Nguifo E. A comparative study on streaming frameworks for big data. In: 44th international conference on very large databases: workshop LADaS—Latin American Data Science, Aug 2018, Rio de Janeiro, Brazil. 2018. p. 1–8.
47. Peng D, Dabek F Large-scale incremental processing using distributed transactions and notifications. In: Proc 9th USENIX conf oper sys. des implement, Vancouver, BC, Canada, 4–6 Oct 2010. 2010. p. 1–15.
48. Marz N. Trident. 2012. https://github.com/nathanmarz/storm/wiki/Trident-tutorial. Accessed 8 Mar 2018.
49. Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues in data stream systems. In: Proc of the 21st ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems (PODS), Madison, Wisconsin, 3–5 June 2002. 2002. p. 1–16.
50. Chandrasekaran S, Cooper O, Deshpande A, Franklin MJ, Hellerstein JM, Hong W, Krishnamurthy S, Madden SR, Reiss F, Shah MA. TelegraphCQ: Continuous dataflow processing. In: Proceedings of the 2003 ACM SIGMOD international conference on management of data, San Diego, California, 9–12 Jun 2003. 2003. p. 668.
51. Abadi DJ, Ahmad Y, Balazinska M, Cherniack M, Hwang JH, Lindner W, Maskey AS, Rasin E, Ryvkina E, Tatbul N, Xing Y, Zdonik S. The design of the borealis stream processing engine. Second biennial conference on innovative data systems research (CIDR 2005). CA: Asilomar; 2005. p. 277–89.
52. Groleat T. High-performance traffic monitoring for network security and management. Human–computer interaction [cs.HC]. Télécom Bretagne; Université de Bretagne Occidentale; 2014.
53. Kamburugamuve S, Fox G, Leake D, Qiu J. Survey of distributed stream processing for large stream sources. Grids UCS Indiana Educ. 2013. https://doi.org/10.13140/rg.2.1.3856.2968.
54. Murthy S. What are the disadvantages of Redis? 2016. https://www.quora.com/What-are-the-disadvantages-of-Redis. Accessed 8 Mar 2018.

55. Su X, Gilman E, Wetz P, Riekki J, Zuo Y, Leppanen T. Stream reasoning for the internet of things: challenges and gap analysis. WIMS '16 proceedings of the 6th international conference on web intelligence, mining and semantics, Nîmes, France—June 13–15, New York: ACM. Article no 1. 2016. https://doi.org/10.1145/2912845.2912853.
56. Morales GDF, Bifet A. SAMOA: scalable advanced massive online analysis. J Mach Learn Res. 2015;16(1):149–53.
57. Amazon Web Services. Lambda architecture for batch and stream processing. 2018. https://d1.awsstatic.com/whitepapers/lambda-architecure-on-for-batch-aws.pdf Accessed 2 May 2019.
58. Kreps J. Questioning the Lambda architecture. 2014. https://www.oreilly.com/ideas/questioning-the-lambda-architecture. Accessed 2 May 2019.
59. Tay Y. Data generation for application-specific benchmarking. Proc VLDB Endowment. 2011;4(12):1470–3.
60. HiBench big data benchmark suite. https://github.com/intel-hadoop/HiBench. Accessed 21 Dec 2018.
61. Hadoop 1.2.1 Documentation. GridMix. https://hadoop.apache.org/docs/r1.2.1/gridmix.html. Accessed 8 Mar 2018.
62. Ouaknine K, Carey M, KirkPatrick S. The PigMix benchmark on Pig, MapReduce, and HPCC systems. 2015 IEEE international conference on big data, New York, NY, USA, 27 June–2 July 2015. p. 643–8. https://doi.org/10.1109/bigdatacongress.2015.99.
63. Ghazal A, Rabl T, Hu M, Raab F, Poess M, Crolotte A, Jacobson H. BigBench: towards an industry standard benchmark for big data analytics. In: Proceedings of the 2013 ACM SIGMOID international conference on management of data, New York, NY, USA, 22–27 Jun 2013. p. 1197–203.
64. Bergamaschi S, Gagliardelli L, Simonini G, Zhu S. BigBench workload executed by using apache flink. Procedia Manuf. 2017;11:695–702. https://doi.org/10.1016/j.promfg.2017.07.169.
65. Wang L, Zhan J, Luo C, Zhu Y, Yang Q, He Y, et al. BigDataBench: a big data benchmark suite from internet services. In: 2014 IEEE 20th international symposium on high performance architecture (HPCA), Orlando, FL, USA: IEEE, 15–19 February 2014. 2014. https://doi.org/10.1109/hpca.2014.6835958.
66. Gao W, Zhan J, Wang L, Luo C, Zheng D, Wen X, et al. BigDataBench: A scalable and unified big data and AI benchmark suite. 2018. arXiv.org > cs > arXiv:1802.08254v2. https://arxiv.org/abs/1802.08254v2.
67. Liao X, Gao Z, Ji W, Wang Y. An enforcement of real-time scheduling in Spark Streaming. 6th international green and sustainable computing conference, IEEE. 2016. https://doi.org/10.1109/igcc.2015.7393730. p. 1–6.
68. Agerri R, Artola X, Beloki Z, Rigau G, Soroa A. Big data for natural language processing: a streaming approach. Knowledge-based systems. 2015;79:36–42 ISSN 0950-7051.
69. Krawczyk B, Woźniak M. Incremental weighted one-class classifier for mining stationary data streams. J Comput Sci. 2015;9:19–25.
70. Chan SWK, Chong MWC. Sentiment analysis in financial texts. Decis Support Syst. 2017;94:53–64.
71. Rakthanmanon T, Campana B, Mueen A, Batista G, Westover B, Zhu Q, Zakaria J, Keogh E. Addressing big data time series: mining trillions of time series subsequences under dynamic time warping. ACM Trans Knowl Discov Data. 2013;7(3):31. https://doi.org/10.1145/2500489.
72. Hadian A, Shahrivari S. High-performance parallel k-means clustering for disk-resident datasets on multi-core CPUs. J Supercomput. 2014;69(2):845–63.
73. Mozafari B, Zeng K, D'Antoni L, Zaniolo C. High-performance complex event processing over hierarchical data. ACM Trans Datab Syst. 2013;38(4):39. https://doi.org/10.1145/2536779.
74. Sun Y, Wang Z, Liu H, Du C, Yuan J. Online ensemble using adaptive windowing for data streams with concept drift. Int J Distrib Sens Netw. Article ID 4218973, 9 pages. 2016. http://dx.doi.org/10.1155/2016/4218973.
75. Nguyen DT, Jung JJ. Real-time event detection on social data stream. Mobile Netw Appl. 2014;20(4):475–86.
76. Tsagkatakis G, Beferull-Lozano B, Tsakalides P. Singular spectrum-based matrix completion for time series recovery and prediction. EURASIP J Adv Signal Proces. 2016;2016:66. https://doi.org/10.1186/s13634-016-0360-0.
77. Papapetrou O, Garofalakis M, Deligiannakis A. Sketching distributed sliding-window data streams. VLDB J. 2015;24:345–68. https://doi.org/10.1007/s00778-015-0380-7.
78. Elkhoukhi H, NaitMalek Y, Berouine A, Bakhouya M, Elouadghiri D, Essaaidi M. Towards a real-time occupancy detection approach for smart buildings. Procedia Comput Sci. 2018;134:114–20.
79. Chakrabarti C. Delivering interactive access to data at massive scale at Barclays. Austin. 2016.
80. Kovacevc I, Mekterovic I. Novel BI data architectures. MIPRO 2018, Opatija, Croatia. 2018. p. 1191–6.
81. Veiga J, Enes J, Exposito RR, Tourino J. BDEv 3.0: energy efficiency and microarchitectural characterization of big data processing frameworks. Fut Generat Comput Syst. 2018;86:565–81.
82. Tozzi, C. Dummy's guide to batch vs. streaming. Trillium Software. 2017. http://blog.syncsort.com/2017/07/big-data/big-data-101-batch-stream-processing/. Accessed 2 Mar 2018.
83. Dusi M, D'Heureuse N, Huici F, Trammell B, Niccolini S. Blockmon: flexible and high performance big data stream analytics platform and its use cases. NEC Tech J. 2012;7:102–6.
84. Puthal D, Nepal S, Ranjan R, Chen J. A dynamic prime number based efficient security mechanism for big sensing data streams. J Comput Syst Sci. 2017;83:22–42.
85. Vanathi R, and Khadir ASA. A robust architectural framework for big data stream computing in personal healthcare real-time analytics. World Congress on Computing and Communication Technologies. 2017. p. 97–104. https://doi.org/10.1109/wccct.2016.32.
86. Ma K, Yang B. Stream-based live entity resolution approach with adaptive duplicate count strategy. Int J Web Grid Serv. 2017;13(3):351–73.
87. Murphy BM, O'Driscoll C, Boylan GB, Lightbody G, Marnane WP. Stream computing for biomedical signal processing: A QRS complex detection case study. In: Conf proc IEEE eng med biol soc. 2015. https://doi.org/10.1109/embc.2015.7319741. p. 5928–31.
88. Apache Spark Streaming—Spark 2.1.0 Documentation. http://spark.apache.org/streaming.
89. Sun H, Birke R, Bjorkqvist M, Chen LY. AccStream: accuracy-aware overload management for stream processing systems. In: IEEE conference on autonomic computing. New York: Elsevier; 2017. p. 39–48.
90. Canbay Y, Sağıroğlu S. Big data anonymization with spark (UBMK'17). In: 2nd IEEE international conference on computer science and engineering. 2017. p. 833–8.

91. Sahana RG, Babu BS. Converting an E-commerce prospect into a customer using streaming analytics. In: 2nd international conference on applied and theoretical computing and communication technology (iCATccT) IEEE. 2016. p. 312–7. https://doi.org/10.1109/icatcct.2016.7912014.
92. Troiano L, Vaccaro A, Vitelli MC. On-line smart grids optimization by case-based reasoning on big data. In: 2016 IEEE workshop on environmental, energy, and structural monitoring systems (EESMS), Bari, Italy, 13–14 Jun 2016.
93. Joseph S, Jasmin EA. Stream computing framework for outage detection in smart grid. In: Proceedings of 2015 IEEE international conference on power, instrumentation, control and computing (PICC), Thrissur, India, 9–11 Dec 2015. 2015. https://doi.org/10.1109/picc.2015.7455744.
94. Apache. Apache Storm. 2016. http://storm.apache.org. Accessed 10 Oct 2018.
95. Gokalp MO, Kocyigit A, Eren PE. A visual programming framework for distributed Internet of Things centric complex event processing. Comput Elect Eng. 2018;74:581–604.
96. Maio CD, Fenza G, Loia E, Orciuoli F. Distributed online temporal fuzzy concept analysis for stream processing in smart cities. J Parallel Distrib Comput. 2017;110:31–41.
97. Val PB, Garcia NF, Sanchez-Fernandez L, Arias-Fisteus J. Patterns for distributed real-time stream processing. IEEE Trans Parallel Distrib Syst. 2017;2(11):3243–57. https://doi.org/10.1109/TPDS.2017.2716929.
98. Fernandez-Rodrigues JY, Alvarez-Garcia JA, Fisteus JA, Luaces MR, Magana VC. Benchmarking real-time vehicle data streaming models for a smart city. Inform Syst. 2017;72:62–76.
99. Bifet A. Mining big data in real time. Informatica (Slovenia). 2013;37:15–20.
100. Apache. Apache Samza-What is Samza? 2016. http://samza.apache.org. Accessed 8 Oct 2018.
101. Ananthanarayanan R, Basker V, Das S, Gupta A, Jiang H, Qiu T, Reznichenko A, Ryabkov D, Singh M, Venkataraman S. Photon: fault-tolerant and scalable joining of continuous data streams. In: Proceedings of 2013 ACM SIGMOD international conference on management of data, New York, New York, USA, 22–27 June 2013. 2013. p. 577–88.
102. Apache Apache Aurora. 2016. http://aurora.apache.org. Accessed 7 Oct 2018.
103. Jiang Q, Adaikkalavan R, Chakravarthy S. MavEStream: synergistic integration of stream and event processing. In: 2007 second international conference on digital telecommunications (ICDT'07) San Jose, CA, USA. 2007. p 29–361. https://doi.org/10.1109/icdt.2007.21 IEEE Xplore.
104. Yang W, Da Silva A, Picard ML. Computing data quality indicators on big data streams using a CEP. In: 2015 International workshop on computational intelligence for multimedia understanding (IWCIM), Prague, Czech Republic, 29–30 Oct 2015. 2015.
105. EsperTech. http://www.espertech.com. Accessed 8 Oct 2018.
106. Song M, Kim MC. RT^{2}M: real-time twitter trend mining system. In: Proceedings of international conference on social intelligence and technology (SOCIETY), State College, PA, USA, 8–10 May 2013. 2013. p. 64–71.
107. Barbieri DF, Braga D, Ceri S. Querying RDF streams with C-SPARQL. ACM Sigmoid. 2010;39(1):20–36. https://doi.org/10.1145/1860702.1860705.
108. Ren X, Khrouf H, Kazi-Aoul Z, ChabChoub Y, Cure O. On measuring performances of C-SPARQL and CQELS. CoRR, abs/1611.08269. 2016.
109. Morales GF. SAMOA: A platform for mining big data streams. WWW 2013 Companions, Rio de Janeiro, Brazil, 13–17 May 2013. 2013.
110. Keeney J, Fallon L, Tai W, O'Sullivan D. Towards composite semantic reasoning for real-time network management data enrichment. In: Proceedings of the 11th international conference on network and service management (CNSM), Barcelona, Spain, 9–13 Nov 2013. 2015. p. 182–6.
111. Le-Phuoc D, Dao-Tran M, Parreira JX, Hauswirth M. A native and adaptive approach for unified processing of linked streams and linked data. In: International semantic web conference, Koblenz, Germany, 23–27 October 2011. 2011. p. 370–88.
112. Anicic D, Rudolph S, Fodor P, Stojanovic N. Stream reasoning and complex event processing in ETALIS. Sem Web Linked Spatiotemp Data Geo-Ontolo. 2012;3(4):397–407.
113. Apache Kylin. Kylin cube from streaming (Kafka). 2015. http://kylin.apache.org/docs15/tutorial/cube_streaming.html. Accessed 2 Oct 2018.
114. Splunk. Splunk Stream. 2017. https://splunkbase.splunk.com/app/1809/. Accessed 2 Oct 2018.
115. Shnayder V, Chen B, Lorincz K, Fulford-Jones TRF, Welsh M. Sensor networks for medical care. Technical report TR-08-05, Division of Engineering and Applied Sciences, Harvard University. 2005. https://www.eecs.harvard.edu/~shnayder/papers/codeblue-techrept05.pdf. Accessed 8 Oct 2018.
116. Dror Y. Practical elastic search anomaly detection made powerful with anodot. 2017. https://www.anodot.com/blog/practical-elasticsearch-anomalydetection-made-owerful-with-anodot/. Accessed 8 Mar 2019.
117. Baciu G, Li C, Wang Y, Zhang X. Cloudets: Cloud-based cognition for large streaming data. In: Ge N, Lu J, Wang Y, Howard N, Chen P, Tao X, Zhang B, Zadeh LA (eds) Proceedings of IEEE 14th international conference on cognitive informatics and cognitive computing (ICCI*CC'15), Tsinghua, Univ., Beijing, China, 6–8 Jul 2015. 2015. p. 333–8.
118. Tedeschi A, Benedetto F. A cloud-based big data sentiment analysis application for enterprises' brand monitoring in social media streams. In: 2015 IEEE 1st international forum on research and technologies for society and industry leveraging a better tomorrow (RTSI), Turing, Italy, 16–18 Sept 2015. 2015. p 186–91.
119. Lavin A, Ahmad S. Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA), Miami, FL, USA, 9–11 Dec 2015. 2015. https://doi.org/10.1109/icmla.2015.141.
120. Chen X, Chen H, Zhang N, Huang J, Zhang W. Large-scale real-time semantic processing framework for Internet of Things. Int J Distrib Sens Netw. 2015;365372:11. https://doi.org/10.1155/2015/365372.
121. Branscombe M. How Microsoft's fast track Azure will help businesses conquer IoT. 2015. http://www.techradar.com/news/internet/cloud-services/howmicrosoft-s-fast-track-azure-will-help-businesses-conquer-iot-1291025. Accessed 8 Mar 2018.
122. Biem A, Bouillet E, Feng H, Ranganathan A, Riabov A, Verscheure O, Koutsopoulos H, Moran C. IBM InfoSphere streams for scalable, real-time, intelligent transportation services. SIGMOID'10 Indianapolis, Indiana, USA, 6–11 Jun 2010. 2010. p. 1093–100.

123. Akidau T, Balikov A, Bekiroglu K, Chernyak S, Haberman J, Lax R, McVeety S, Mills D, Nordstrom P, Whittle S. MillWheel: fault-tolerant stream processing at internet scale. Proc VLDB Endowment. 2013;6(11):1033–44.
124. Blount M, Ebling MR, Eklund JM, James AG, McGregor C, Percival N, Smith KP, Sow D. Real-time analysis for intensive care: development and deployment of the artemis analytic system. IEEE Eng Med Biol Mag. 2010;29(2):110–8. https://doi.org/10.1109/MEMB.2010.936454.
125. Introducing WSO2 Data Analytics Server. 2015. https://docs.wso2.com/display/DAS300/Introducing+DAS. Accessed 8 Mar 2019.
126. Ali M, Chandramouli B, Goldstein J, Schindlauer R. The extensibility framework in Microsoft StreamInsight. In: Proceedings of the 2011 IEEE 27th international conference on data engineering (ICDE), Washington, DC, USA, 11–16 Apr 2011. 2011. p. 1242–53.
127. TIBCO StreamBase Documentation. https://docs.tibco.com. Accessed 8 Mar 2018.
128. Wilkes S. Making in-memory computing enterprise-grade—overview–Striim. 2016. http://www.striim.com/blog/2016/06/making-in-memorycomputing-enterprise-grade-overview/ Accessed 8 Mar 2019.
129. Kyvos Insights. Kyvos insights 2018. 2018. https://www.kyvosinsights.com/. Accessed 1 Feb 2018.
130. AtScale. AtScale overview (version 4.1). 2017. http://info.atscale.com/atscale-overview. Accessed 2 Feb 2018.
131. AtScale. AtScale. 2018. http://atscale.com/product/. Accessed 2 Feb 2018.
132. Gedik B, Andrade H, Wu K, Yu PS, Doo M. Spade: the S declarative stream processing engine. In: 2008 ACM SIGMOID international conference on management of data, Vancouver, Canada, 9–12 Jun 2008. 2008. p. 1123–34.
133. Mimic, II. http://physionet.org/physiobank/database/mimic2db/. Accessed 4 Nov 2016.
134. Wu Z, Zou M. An incremental community detection method for social tagging systems using locality sensitive hashing. Neural Netw. 2014;58:14–28. https://doi.org/10.1016/j.neunet.2014.05.019.
135. O'Callaghan L, Mishra N, Meyerson A, Guha S, Motwani R. Streaming-data algorithms for high-quality clustering. In: Proceedings of IEEE international conference on data engineering, San Jose, CA, USA, 26 Feb–1 Mar 2002. 2002. p. 685–94.
136. Aggarwal CC, Han JW, Wang JY. A framework for clustering evolving data streams. In: Proceedings of the 29th VLDB conference, vol. 29, Berlin, Germany, 9–12 Sep 2003. 2003. p. 81–92.
137. Backhoff O, Ntoutsi E. Scalable online-offline stream clustering in apache spark. In: 2016 IEEE 16th international conference on data mining workshops (ICDMW), Barcelona, Spain, 12–15 Dec 2016. 2016. p. 37–44. https://doi.org/10.1109/icdmw.2016.0014.
138. Aggarwal CC, Han J, Wang J, Yu PS. A framework for projected clustering of high dimensional data streams. In: Proceedings of the 30th international conference on very large data bases, 30, Toronto, Canada, 31 Aug–3 Sep 2004. 2004. p. 852–63.
139. Cao F, Ester M, Qian W, Zhou A. Density-based clustering over an evolving data stream with noise. In: 2006 SIAM conference on data mining. 2006. p. 328–39.
140. Chen Y, Tu L. Density-based clustering for real-time stream data. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, CA, USA, 12–15 Aug 2007. 2007. p. 133–42.
141. Zhu WH, Yin J, Xie YH. Arbitrary shape cluster algorithm for clustering data stream. J Softw. 2006;17(3):379–87.
142. Khalilian M, Mustapha N, Sulaiman N. Data stream clustering by divide and conquer approach based on vector model. J Big Data. 2016;3:1. https://doi.org/10.1186/s40537-015-0036-x.
143. Dai DB, Zhao G, Sun SL. Effective clustering algorithm for probabilistic data stream. J Softw. 2009;20(5):1313–28.
144. Ding S, Zhang J, Jia H, Qian J. An adaptive density data stream clustering algorithm. Cogn Comput. 2016;8(1):1–9. https://doi.org/10.1007/s12559-015-9342-z.
145. Choi D, Song S, Kim B, Bae I. Processing moving objects and traffic events based on spark streaming. In: Proceedings of the 8th international conference on disaster recovery and business continuity (DRBC), Jeju, South Korea, 25–28 Nov 2015. 2015. p. 4–7.
146. Chen XJ, Ke J. Fast processing of conversion time data flow in cloud computing via weighted FPtree mining algorithms. In: 2015 IEEE 12th intl conf on ubiquitous intelligence and computing and 2015 IEEE 12th intl conf on autonomic and trusted computing and 2015 IEEE 15th intl conf on scalable computing and communications and its associated workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 Aug 2015. 2015.
147. Li T, Wang L. Key technology of online auditing data stream processing. In: 2015 IEEE 12th intl conf on ubiquitous intelligence and computing and 2015 IEEE 12th intl conf on autonomic and trusted computing and 2015 IEEE 15th intl conf on scalable computing and communications and its associated workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 Aug 2015. 2015.
148. Xiao F, Aritsugi M, Wang Q, Zhang R. Efficient processing of multiple nested event pattern queries over multi-dimensional event streams based on a triaxial hierarchical model. Artif Intell Med. 2016;72(1):56–71. https://doi.org/10.1016/j.artmed.2016.08.002.
149. Wang Z, Zhao Z, Weng S, Zhang C. Incremental multiple instance outlier detection. Neural Comput Appl. 2015;26:957–68. https://doi.org/10.1007/s00521-014-1750-6.
150. Ruiz E, Casillas J. Adaptive fuzzy partitions for evolving association rules in big data stream. Int J Approx Reasoning. 2018;93:463–86.
151. Jadhav SA, Kosbatwar SP. Concept-adapting very fast decision tree with misclassification error. Int J Adv Res Comput Eng Technol (IJARCET). 2016;5(6):1763–7.

Publisher's Note