Challenges and Solutions for Processing Real-Time Big Data Stream: A Systematic Literature Review

ERUM MEHMOOD AND TAYYABA ANEES
School of Systems and Technology, University of Management and Technology, Lahore 54770, Pakistan
Corresponding author: Erum Mehmood (erum9964@hotmail.com)

ABSTRACT

Recent real-time data warehousing (DWH) and big data streaming have become ubiquitous due to the fact that a number of business organizations are gearing up to gain competitive advantage. The capability of organizing big data in efficient manner to reach a business decision empowers data warehousing in terms of real-time stream processing. A systematic literature review for real-time stream processing systems is presented in this paper which rigorously look at the recent developments and challenges of real-time stream processing systems and can serve as a guide for the implementation of real-time stream processing framework for all shapes of data streams. Background: Published surveys and reviews either cover papers focusing on stream analysis in applications other than real-time DWH or focusing on extraction, transformation, loading (ETL) challenges for traditional DWH. This systematic review attempts to answer four specific research questions. Research Questions: 1) Which are the relevant publication channels for real-time stream processing research? 2) Which challenges have been faced during implementation of real-time stream processing? 3) Which approaches/tools have been reported to address challenges introduced at ETL stage while processing real-time stream for real-time DWH? 4) What evidence have been reported while addressing different challenges for processing real-time stream? Methodology: A systematic literature was conducted to compile studies related to publication channels targeting real-time stream processing/joins challenges and developments. Following a formal protocol, semi-automatic and manual searches were performed for work from 2011 to 2020 excluding research in traditional data warehousing. Of 679,547 papers selected for data extraction, 74 were retained after quality assessment. Findings: This systematic literature highlights implementation challenges along with developed approaches for real-time DWH and big data stream processing systems and provides their comparisons. This study found that there exists various algorithms for implementing real-time join processing at ETL stage for structured data whereas less work for un-structured data is found in this subject matter.

INDEX TERMS
Real-time stream processing, big data streaming, structured/un-structured data, ETL, systematic literature review.

I. INTRODUCTION

Real-time analytics are becoming ubiquitous for several application scenarios where well-timed business decisions are extremely important. Processing continuous and big data streams for real-time analytics is very challenging while implementing ETL stage for data warehousing (DWH) or other big data applications due to the nature of big data with respect to volume, variety, velocity, volatility, variability, veracity, and value [1], [2].

Continuous supply of big data is referred as stream. Stream of data may be generated from single or multiple big data sources shown in figure 1. A broad category of applications participate in continuous generation of massive data. Analysis of these streams is a big challenge where gathered data is heterogeneous and can be of any shape/nature i.e., structured, semi/unstructured, symmetrical or skewed. Big portion of massive data resulted from real-time stream need real-time processing/analysis as value of data considered in its freshness. Real-time stream processing (refer as in-memory processing of massive data) can be generally required into two types of application domains: first where organising data

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao.
to reach a decision is required (real-time DWH) and second where to generate a certain reaction on real-time basis is essential particularly with low latency. Few applications of second type are also listed in figure 1. Before being loaded into these applications, streams need to be processed ensuring data quality. This necessitates real-time stream processing/analysis.

Several operations are required for stream processing which include data cleaning, query processing, stream-stream join, stream-disk join, data transformation etc. A broad category of approaches, tools and technologies have been developed so far to overcome the challenges for stream processing. These approaches possibly deal with multiple shapes and storage models of data, applying several operations on these streams.

To address real-time stream processing challenges various approaches have been developed so far: stream-stream join algorithms, stream-disk join algorithms with reduced data latency/skewed data, distributed streaming ETL, Mesa DWH, streaming processing framework, distributed join processing, sensor networks, object tracking and monitoring, and multi-join query processing in cloud DWH. In addition, processed output must be provided with low latency, limited resources, accuracy and within seconds to make real-time reactions and decisions possible. The depth of challenges targeting real-time stream processing has produced so much research that it is required to conduct a systematic analysis of proposed solutions.

The focus of this study is to present an extensive systematic literature review (SLR) to gather different approaches for real-time stream processing for all possible application domains specifically real-time DWH. We have finalized 74 studies out of 667,414 total papers for this review based on quality assessment criteria. The novelty of our SLR is that it provides a new classification criteria, real-time stream processing research targeting channels, real-time DWH/big data streaming challenges, approaches to address these challenges after validating studies empirically.

This paper is organized as follows: Existing reviews related to stream processing are presented in section II. Research methodology followed to conduct this survey is discussed in section III including the objectives, quality assessment criteria and research questions. Assessment and discussion of research questions are demonstrated in section IV. Concluded discussion and future directions are presented in Section V.

II. RELATED WORK
It was found that most of the existing surveys and systematic reviews do not cover publication channels approaches, challenges and solutions targeting real-time stream processing research needed in business intelligence, and focus majorly on tools used for big data analytics and DWH design approaches from social media. A recent systematic literature review of big data stream analysis is presented by [6]. Authors have reviewed key issues for big data stream analysis and tools/technologies employed to address these issues. However, this study has not focused on research and challenges in real-time stream analysis and real-time DWH domains.

Authors in [3] presented a review on fundamental use of big data analytics in various businesses/industries in terms of helping and maintaining their resources, using Scopus digital repository for searching relevant articles. Big data technologies/platforms, their services, applications, programming languages, data aggregation tools, databases and DWHs have also been reviewed in this study. This study enlisted platforms...
for examining big data sets, both un-structured and structured for big data analytics which are: Hadoop, GridGain, MapReduce, HPCC and Apache Storm. They also highlighted tools for database/DWH as: Cassandra, MongoDB, CouchDB, Terrastore, Hibari, Hypertable, Hive, Infinispan, HBase, Neo4j, OrientDB, etc. However, this study does not focus on challenges and approaches developed in the field of real-time DWH and big data streaming.

Likewise, [4] conducted a study that is centered on competitive analysis of social media data and transformation into knowledge and on DWH design approaches from social media. As social media considered as massive dynamic and un-structured data, making them more challenging for companies to use, analyze and store these data. This study classified DWH design approaches from social media into two heads: incorporation of sentiment analysis in DWH schema and behaviour analysis. Nevertheless, focus of this study is not on challenges and approaches developed in the field of real-time DWH and big data streaming.

TABLE 1. Comparison with related works.

| Paper | Focus of Survey | Newest Ref. | Survey Approach | Quality Assessment Score | Aspects of real-time stream processing: | Evidences | Targeted Digital Repositories |
|-------|----------------|-------------|----------------|--------------------------|--------------------------------------|-----------|-----------------------------|
| [3]   | To analyze theoretical contributions and tools used for big data analytics focusing social media exploration, text mining and machine learning applications | 2017 | Informal | x | x | x | 1 |
| [4]   | To highlight data warehouse design approaches from social media focusing sentiment analysis in DWH schema | 2017 | Informal | x | x | x | Not mentioned |
| [5]   | To identify and compare applications of stream analytics for different use cases based on partitioning, state management and fault tolerance criteria | 2019 | Informal | x | x | x | Not mentioned |
| [6]   | To review big data streaming tools and technologies employed for stream analysis | 2019 | Systematic search | x | x | x | 4 |
| This survey | To identify and compare challenges and approaches for processing real-time big data stream by applying a strict review protocol and methodical approach | 2020 | Systematic search, snowballing and assessment | ✓ | ✓ | ✓ | ✓ | 10 |

Our review distinguishes itself from the above reviews by focusing on the publication channels in real-time stream processing, big data streaming, closely examining the ETL stage for real-time DWH. In addition, we employ a more rigorous approach than all of the above reviews by following strict criteria and quality assessment scoring. Table 1 gives aspect wise comparison of existing surveys with our survey. There seem to be no existing survey that cover all features related to real-time stream processing as well as not considering real-time DWH literature.

III. RESEARCH METHODOLOGY

Guidelines for systematic reviews provided in software engineering research by [7] and [8] are followed by our survey. According to these guidelines, we have included three main phases in our research methodology: plan, conduct and report of review.

A. REVIEW PLAN

Figure 2 shows the research methodology, which demonstrates search process for relevant research activities, definition of a categorization scheme, and mapping of articles.
TABLE 2. Research Questions (RQs).

| (RQ) | Research Question Statement | Motivation |
|------|----------------------------|------------|
| RQ 1 | Which are the relevant publication channels for real-time stream processing research? | To identify where real-time stream processing research can be found as well as the good targets for publication of future studies. To assess the quality of targeted publication channels based on their ranking. To categorize the publications for the selected time period, channel types and targeting geographical areas. Moreover, we aim to classify publications according to their research types, used approaches and empirical type. |
| RQ 1.1 | How many articles have been published in between Jan 2011 till Jan 2020? | |
| RQ 1.2 | Which channel types and geographical areas targeting real-time stream processing research? | |
| RQ 1.3 | What are the research types, approaches and applications of selected studies? | |
| RQ 1.4 | How many of these studies have validated their approaches empirically? | |
| RQ 2 | Which challenges have been faced during implementation of real-time stream processing? | |
| RQ 2.1 | What tools/technologies/approaches have been developed to address these challenges? | To evolve and streamline real-time stream processing with different requirements and to encourage different perspectives for future research. To identify application domains where real-time stream processing are required other than DWII. We aim to distinguish among challenges faced by applications particularly IoT and Social Media. |
| RQ 2.2 | Which developments in the domain of IoT and Social Media have been reported while processing real-time data streams? | |
| RQ 3 | Which approaches/tools have been reported to address challenges introduced at ETL stage while processing real-time stream for real-time DWII? | To identify the existing stream/semi-stream-structured/un-structured data join approaches reported in the existing real-time stream processing literature to address challenges identified in implementing real-time DWII. We aim to identify methodology adopted by studies along with listing of data structure used for particular shape of data to assess the effect of that methodology. |
| RQ 3.1 | Which shape and structure of data used by these studies for implementing their approach? | |
| RQ 4 | What evidence have been reported while addressing different challenges for processing real-time stream? | To assess the evidence provided by these studies which lead to significantly more accurate approach. To identify datasets used while experimenting proposed approach in these studies (synthetic or real-life). To identify the impact of approaches on performance and cost, and to assess the frequency for adoption of specific performance benchmark for validation. |
| RQ 4.1 | Which datasets have been used for experiments by these approaches? | |
| RQ 4.2 | What performance benchmarks have been adopted by these approaches for experiments using real-life datasets? | |

FIGURE 2. Research methodology.

A highly structured process has been followed in this review that involved:

- Research objectives
- Specifying research questions (RQs)
- Organizing searches of databases
- Studies selection
- Screening relevant studies
- Data extraction
- Results synthesising
- Finalizing the review report

1) RESEARCH OBJECTIVES
Core objectives of our research are as follows:

a) To develop a library of articles related to developments of real-time stream processing during ETL phase or others, and make this dataset available to other researchers.

b) To identify more significant work that provides direction to investigate challenges for real-time stream processing, ETL and real-time DWII.

c) To distinguish research gaps for ETL, real-time stream processing and DWII in recent studies.

d) Characterise existing approaches and solutions for the challenges while implementing real-time stream processing for heterogeneous, structured/unstructured data and clarify the similarities and differences between them.

2) RESEARCH QUESTIONS
It is important to formulate the primary RQs in order to conduct this SLR. These RQs are developed to identify relevant publication channels, challenges/developments and evidences of approaches for real-time stream processing systems as mentioned in table 2.

B. REVIEW CONDUCT
To strengthen the reliability and validity for our search results two reviewers participated for the inclusion of papers. The process of conducting this review has been articulated in four steps presented below. In first step, relevant primary studies have been searched from most commonly used digital libraries. Selection of studies based on pre-defined inclusion/exclusion criteria has been performed during second
FIGURE 3. Search keywords used to identify works included in our knowledge base.

step. We have designed quality assessment criteria to further enhance quality of our review described in third step. Backward snowballing is then performed to extract important candidate papers during final fourth step.

1) SEMI-AUTOMATED SEARCH IN DIGITAL LIBRARIES
A systematic research has been carried out to filter irrelevant studies and extract appropriate information. Therefore, semi-automatic and manual search techniques have been followed while exploring the search terms. Semi-automated search has been conducted in seven digital libraries mentioned below:

- ACM Digital Library [http://dl.acm.org]
- IEEE eXplore [http://ieeexplore.ieee.org]
- ScienceDirect [https://www.sciencedirect.com]
- SpringerLink [https://link.springer.com]
- IGI Global [https://www.igi-global.com/search/]
- Inderscience Online [https://www.inderscienceonline.com/]
- Hindawi [https://www.hindawi.com/]
- MDPI (Multidisciplinary Digital Publishing Institute) [https://www.mdpi.com/]
- arXiv [https://arxiv.org/search/cs]
- Taylor & Francis Online [https://www.tandfonline.com/]

Apart from this, some other digital libraries were also explored but not included due to accessibility constraints. The objective of manual search is to collect more literature relevant to real-time stream processing and DWH. Extracted information can be more relevant for limited search terms therefore following conditions were applied to limit our search terms:

- Based on formulated RQs, determine primary keywords.
- Identification of secondary keywords and synonyms for additional keywords.
- ‘AND’ and ‘OR’ Boolean operators have been incorporated with keywords to develop a search string.

Possible arrangements of search string used can be noted from figure 3. Primary keywords were selected as key identifiers for research of real-time stream processing. Primary keywords were chosen along with any of secondary or additional keywords. Combination of keywords, Boolean operators and wildcard have developed a final search string mentioned as:

(real time OR real time data warehousing*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analy* OR problem*)

Table 3 demonstrates the final search strings used to search the seven digital libraries. Semi-automatic search was limited to only titles for ACM journals, IEEExplore and ScienceDirect. Due to limit of five wildcard characters for search in IEEExplore, search string needed to be slightly changed for this library. Irrelevant hits were reduced due to this setting. Other digital libraries were explored with “all fields” setting, as these do not allow a more specific search configuration. Search string being too restrictive failed to find relevant articles for IGI Global digital library, therefore final search string designed for this library contains less number of keywords shown in table 3. Final search string failed when applied for digital library Hindawi, therefore search was conducted with few keywords resulted in some relevant hits.
E. Mehmood, T. Anees: Challenges and Solutions for Processing Real-Time Big Data Stream

TABLE 3. Search strategies for digital libraries.

| Digital Library | Search String | Applied Filters |
|-----------------|---------------|-----------------|
| ACM digital library | [[All: real time] OR [All: real time data warehouse*]] AND [[All: stream] OR [All: semi-stream] OR [All: etl] OR [All: challenges] OR [All: extract*,transform*,load*]] AND [[All: join] OR [All: process*] OR [All: analy*] OR [All: problem*]] | [01/01/2011 TO 01/31/2020] |
| IEEEExplore | (real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analytics OR problems) | 2011-Jan2020 |
| ScienceDirect | (real time OR real time data warehouse) AND (stream OR semi-stream OR ETL OR challenges OR (extraction,transformation,loading)) AND (join OR process OR analysis OR problems) | 2011-Jan2020 |
| SpringerLink | ‘(real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analy* OR problem*)’ | Computer Science, (2011-Jan2020) |
| IGI Global | (real time) AND (data warehousing or stream processing) | Individual Journal Articles (2011-Jan2020) |
| Inderscience Online | (real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analytics OR problems) | 2011-Jan2020 |
| Hindawi | “real time stream processing OR real time data warehouse” | 2011-Jan2020 |
| MDPI | “real time stream processing” | 2011-Jan2020 |
| arXiv | “real time data warehouse OR real-time stream processing” | 2011-Jan2020 |
| Taylor & Francis Online | (real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analy* OR problem*) | 2011-Jan2020 |

TABLE 4. Possible ratings for recognized and stable publication source.

| Publication Source                          | Scores |
|--------------------------------------------|--------|
|                                            | 4  | 3  | 2  | 1  | 0  |
| Journals                                   | Q1 | Q2 | Q3 | Q4 | No JCR Ranking |
| Conferences, Workshops, Symposia           | CORE A* | CORE A | CORE B | CORE C | Not in CORE Ranking |

2) SELECTION BASED ON INCLUSION/EXCLUSION CRITERIA

1) Inclusion criteria:
   a) Papers included in review must be in the domain of real-time stream processing.
   b) Papers must target RQs.
   c) Papers published in journals, conferences or workshops are included in the review.
   d) Papers discussing developments and applications of real-time stream processing.

2) Exclusion criteria:
   a) Remove papers written in non-english.
   b) Remove papers that do not discuss real-time stream processing in DWH or big data domain.
   c) Remove the papers published before 2011.
   d) Remove papers discussing simulation domains or traditional DWH.
   e) Remove papers that were written by same research group with same data (most recent was kept in this case).

3) SELECTION BASED ON QUALITY ASSESSMENT

Selection of relevant studies on the basis of quality assessment (QA) is considered as most important step for conducting any review. As the primary studies vary in design therefore quantitative, qualitative, and mixed-method critical appraisal tool used by [9] and [10] are followed to perform QA in our review. In order to enhance our study, we have carried out QA by designing a questionnaire to evaluate the quality of selected articles. The QA of our study was conducted by two reviewers and each study is scored based on the following criteria:

a) The study has awarded score (1) if it contributes towards real-time stream processing or continuous data loading in DWH, otherwise scored (0).

b) If clear solutions to the challenges for implementation of real-time stream processing or DWH have been provided by the study: “Yes (2)”, “Limited (1)”, and “No (0)” were the possible scores.

c) Score (1) is awarded to studies which presents empirical results otherwise scored (0).

d) By taking computer science conference rankings [11], and the journal and country ranking lists [12] into account, the studies were rated. Possible scores for publications from recognized and stable sources are shown in table 4.

A final score has been calculated for each study after adding scores of above questions: (an integer between 0 to 8).
TABLE 5. Selection phases and results.

| Phase | Selection criteria | ACM digital library | IEEEExplore | ScienceDirect | SpringerLink | IGI Global | InDexScience | Hindawi | MDPI | arXiv | Taylor & Francis Online | Total Papers |
|-------|--------------------|---------------------|-------------|---------------|-------------|------------|--------------|----------|------|-------|------------------------|--------------|
| 1     | Search             | 161786              | 16923       | 345652        | 5204        | 154        | 1443         | 135827   | 6    | 305   | 114                    | 667414       |
| 2     | Screening          | 5994                | 147         | 987           | 215         | 65         | 89           | 34       | 10   | 6     | 3625                   | 7552         |
| 3     | Screening          | 3042                | 78          | 98            | 122         | 45         | 63           | 16       | 4    | 3     | 4                      | 3625         |
| 4     | Screening          | 701                 | 34          | 14            | 56          | 8          | 12           | 9        | 4    | 1     | 2                      | 841          |
| 5     | Inspection         | 8                   | 20          | 5             | 24          | 3          | 3            | 4        | 4    | 1     | 2                      | 74           |

Articles achieving scores 3 or more have been included in finalized results.

4) SELECTION BASED ON SNOWBALLING
After performing quality assessment, we conducted backward snowballing [13] through reference list of each finalized study to extract papers. Only those important candidate papers are selected which passed through inclusion/exclusion criteria. Once the paper is found, inclusion/exclusion of that paper has been decided after reading its abstract and then other parts of paper. After having examined selected papers thoroughly we identified one more study [5], and totally added up to 74 primary studies.

C. REVIEW REPORT
Overview of selected studies is provided in this section.

1) OVERVIEW OF THE INTERMEDIATE SELECTION PROCESS OUTCOME
ETL challenges, real-time stream processing and DWH are correlated and extremely active fields in business intelligence, therefore our review methodology had to empirically and systematically draw relevant studies from all related digital libraries. The next stage of our systematic review is to select the papers that will form the knowledge base for this review. About 667,414 papers are left after removing papers older than year 2011.

After building a knowledge base from seven digital publishers in computer science, authors examined title, abstract, and the corresponding full paper if required of each search result. Papers less than four pages long and irrelevant papers were eliminated in this process.

To ascertain the relevance and contribution, accepted publications have been read thoroughly during inspection phase.

2) OVERVIEW OF THE SELECTED STUDIES
Significant results of primary search, filtering and inspection phases, covering ten digital libraries, are presented in table 5. The search resulted in a very big number of papers (667414) while filtering/inspection phases helped reduce this number to 74 articles.

IV. ASSESSMENT AND DISCUSSION OF RESEARCH QUESTIONS
This section concludes the results of our study and provides the descriptive evaluation of each study in tabular format. We analyzed 74 finalized studies based on our RQs in this section.

A. ASSESSMENT OF RQ1: WHICH ARE THE RELEVANT PUBLICATION CHANNELS FOR REAL-TIME STREAM PROCESSING RESEARCH?
Analysis of existing developments and challenges for real-time stream processing is a key challenge for researchers for the development of business intelligence technologies. For this purpose, identification of high quality publication venues and scientometric analysis based on meta information in the area of real-time stream processing is required. In this section, an insightful knowledge of publication sources, types, year and geographical distribution, publication channel wise distribution of selected studies for the evaluation of real-time stream processing research is presented.
Selected paper count each year is shown in figure 4. Note that highest number of selected papers were published last year indicating growing need of research in the field of real-time stream processing and DWH. Figure 5 shows percentage of studies selected from journals, conferences or workshop. Journal publications are generally considered superior specially with a high impact factor, therefore, we have included 56% journal publications in our SLR, all published in Q1-Q4 quartile journals. On the other hand, conference articles are as valuable as journal publications in terms of measuring the performance of a scientific publications, therefore, 42% of selected studies are from good ranked conference articles. Figure 6 presents percentage of geographical distribution of selected research papers. Researchers from Asia and New Zealand contributed most towards developments in real-time stream processing indicating increasing need to shorten the time lag between data acquisition and decision making in these regions.

The overall quality assessment score of finalized studies with detail of overall classification result is mentioned in table 6. Selected papers were classified based on four factors: research type, empirically validated, applied approach and application of study. We have categorized types of research as: SLR, solution proposal, evaluation research or experience paper. It is calculated from table 6 that 97% of selected studies score more that 3 and 82% of final studies have empirically validated their approaches through experiments awarded score 1 shown in category (c) of quality assessment criteria. Studies score less than 3 have been excluded from this SLR.

Major application domains identified during the analysis of selected studies are: real-time DWH, streaming big data for social media and sensor networks, distributed join and stream processing and real-time stream processing. In addition to that, only eighteen papers out of seventy four score zero for category (d) of quality assessment criteria showing unstable/unrecognized publication sources, rest of them score higher indicating competent sources. Due to the relevancy of these eighteen studies, we have included them in our survey. These studies appeared in making important contribution to the area domain.

Table 7 highlights all the publication sources/channels, number of articles per publication source and their percentage contribution towards this study. It is noted that articles related to real-time stream processing applications and techniques have not been published in any particular sources. Along with domain specific sources, various open access sources also welcome stream processing related articles. About 5% of finalized studies have been published in Q1 ranked journal “IEEE Access” and another 5% in “International Conference on Digital Information Management” conference.

B. ASSESSMENT OF RQ2: WHICH CHALLENGES HAVE BEEN FACED DURING IMPLEMENTATION OF REAL-TIME STREAM PROCESSING?

In this section, an insightful knowledge of the issues while implementing real-time stream processing is presented. Two major applications domains of real-time stream processing are categorized as: ETL/real-time DWH (will be discussed during assessment of RQ3), and applications other
TABLE 6. Classification.

| References | Classification | Quality Assessment |
|------------|----------------|--------------------|
| [3] | Journal 2017 SLR | Social media | 1 | 2 | 0 | 4 | 7 |
| [4] | Journal 2017 Evaluation research | Survey | Guideline | ETL | 1 | 1 | 1 | 4 | 7 |
| [5] | Journal 2019 Review | No | Survey | Stream processing framework | 1 | 2 | 0 | 4 | 7 |
| [6] | Journal 2019 SLR | No | Formal | Big data streaming tools | 1 | 1 | 0 | 4 | 6 |
| [14] | Conference 2011 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 4 | 8 |
| [15] | Conference 2017 Evaluation research | Survey | Guideline | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [16] | Conference 2015 Solution proposal | Experiment | Framework | Real-time DWH | 1 | 2 | 1 | 1 | 5 |
| [17] | Journal 2011 Solution proposal | Experiment | Framework | Sensor networks, object tracking & monitoring, etc | 1 | 0 | 1 | 4 | 6 |
| [18] | Journal 2019 Solution proposal | Experiment | Algorithm | Distributed join processing | 1 | 2 | 1 | 4 | 8 |
| [19] | Conference 2011 Review | No | Guideline | Distributed stream processing | 1 | 2 | 0 | 4 | 7 |
| [20] | Conference 2017 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [21] | Conference 2013 Evaluation research | No | Method | Real-time DWH | 1 | 2 | 0 | 0 | 3 |
| [22] | Conference 2012 Solution proposal | Experiment | Framework | Distributed real-time DWH | 1 | 2 | 1 | 2 | 6 |
| [23] | Conference 2015 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [24] | Journal 2019 Solution proposal | Experiment | Module | Real-time DWH | 1 | 2 | 1 | 4 | 8 |
| [25] | Conference 2015 Review | No | Guideline | Real-time DWH | 1 | 2 | 0 | 0 | 3 |
| [26] | Conference 2013 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [27] | Conference 2011 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [28] | Conference 2011 Solution proposal | Experiment | Tool | Real-time DWH | 1 | 2 | 1 | 2 | 6 |
| [29] | Conference 2018 Solution proposal | No | Framework | Real-time DWH | 1 | 2 | 0 | 0 | 3 |
| [30] | Conference 2017 Review | Survey | Guideline | ETL | 1 | 2 | 1 | 0 | 4 |
| [31] | Conference 2015 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 1 | 5 |
| [32] | Conference 2014 Evaluation research | Experiment | Method | Mesa DWH | 1 | 2 | 1 | 0 | 4 |
| [33] | Conference 2015 Evaluation research | Experiment | Guideline | Multiple | 1 | 2 | 1 | 0 | 4 |
| [34] | Conference 2012 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [35] | Conference 2018 Evaluation research | Experiment | Method | Real-time ETL | 1 | 2 | 1 | 0 | 4 |
| [36] | Conference 2012 Evaluation research | Experiment | Method | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [37] | Journal 2016 Solution proposal | Experiment | Method | Mesa DWH | 1 | 2 | 1 | 4 | 8 |
| [38] | Journal 2011 Review | No | Guideline | Multiple | 1 | 0 | 0 | 4 | 5 |
| [39] | Conference 2019 Review | Survey | Guideline | ETL | 1 | 1 | 1 | 2 | 5 |
| [40] | Conference 2016 Solution proposal | Experiment | Module | ETL | 1 | 2 | 1 | 0 | 4 |
| [41] | Journal 2013 Experience paper | Experiment | Algorithm | ETL | 0 | 1 | 1 | 4 | 6 |
| [42] | Journal 2019 Review | No | Guideline | Real-time data streams | 1 | 1 | 0 | 4 | 6 |
| [43] | Journal 2017 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 4 | 8 |
| [44] | Journal 2018 Solution proposal | Experiment | Frame work | Real-time DWH | 1 | 2 | 1 | 3 | 7 |
| [45] | Journal 2014 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 3 | 7 |
| [46] | Conference 2014 Solution proposal | Experiment | Tool | Real-time DWH | 1 | 2 | 1 | 0 | 4 |
| [47] | Conference 2011 Solution proposal | Experiment | Algorithm | Real-time DWH | 1 | 2 | 1 | 2 | 6 |
TABLE 6. (Continued.) Classification.

| References | Classification   | Quality Assessment |
|------------|------------------|--------------------|
|            | References       | Score              |
|            | P.Channel        | (a)                |
| [48]       | Conference 2012  | 1                  |
| [49]       | Conference 2013  | 1                  |
| [50]       | Journal 2018     | 1                  |
| [51]       | Journal 2019     | 1                  |
| [52]       | Journal 2019     | 1                  |
| [53]       | Journal 2016     | 1                  |
| [54]       | Journal 2019     | 1                  |
| [55]       | Journal 2017     | 1                  |
| [56]       | Journal 2014     | 1                  |
| [57]       | Journal 2018     | 1                  |
| [58]       | Journal 2019     | 1                  |
| [59]       | Journal 2018     | 1                  |
| [60]       | Journal 2017     | 1                  |
| [61]       | Journal 2017     | 1                  |
| [62]       | Journal 2015     | 1                  |
| [63]       | Conference 2014  | 1                  |
| [64]       | Conference 2013  | 1                  |
| [65]       | Journal 2019     | 1                  |
| [66]       | Journal 2018     | 1                  |
| [67]       | Journal 2019     | 1                  |
| [68]       | Journal 2013     | 1                  |
| [69]       | Journal 2020     | 1                  |
| [70]       | Journal 2016     | 1                  |
| [71]       | Conference 2017  | 1                  |
| [72]       | Journal 2015     | 1                  |
| [73]       | Journal 2016     | 1                  |
| [74]       | Journal 2017     | 1                  |
| [75]       | Journal 2014     | 1                  |
| [76]       | Journal 2018     | 1                  |
| [77]       | Journal 2015     | 1                  |
| [78]       | Journal 2019     | 1                  |
| [79]       | Journal 2019     | 1                  |
| [80]       | Journal 2018     | 1                  |
| [81]       | Journal 2018     | 1                  |
| [82]       | Journal 2014     | 1                  |

than DWH. Discussion on requirements/challenges and developments for real-time stream processing applications other than real-time ETL are presented separately in subsequent subsections:

1) REQUIREMENTS/CHALLENGES FOR STREAMING BIG DATA IN APPLICATIONS OTHER THAN DWH

Various studies have highlighted and addressed many requirements and challenges in the field of streaming big data and
| Publication source                                           | Channel            | References | No. | %age |
|-------------------------------------------------------------|--------------------|------------|-----|------|
| IEEE Transactions on Knowledge and Data Engineering         | Journal            | [17]       | 1   | 2    |
| IEEE Access                                                 | Journal            | [18], [24], [5] | 3   | 5    |
| IEEE Transactions on Big Data                               | Journal            | [19]       | 1   | 2    |
| Communications of the ACM                                   | Journal            | [37], [38] | 2   | 3    |
| International Journal of Data Warehousing and Mining         | Journal            | [70]       | 1   | 2    |
| Information Systems                                         | Journal            | [41], [43] | 2   | 3    |
| Future Generation Computer Systems                           | Journal            | [42]       | 1   | 2    |
| The Journal of Supercomputing                               | Journal            | [44], [62] | 2   | 3    |
| Knowledge and Information Systems                            | Journal            | [45]       | 1   | 2    |
| Distributed and Parallel Databases                          | Journal            | [50], [59] | 2   | 3    |
| Journal of Big Data                                         | Journal            | [6], [51]  | 2   | 3    |
| Innovations in Systems and Software Engineering              | Journal            | [52]       | 1   | 2    |
| Journal of Signal Processing                                | Journal            | [53]       | 1   | 2    |
| Science China Information Sciences                           | Journal            | [54]       | 1   | 2    |
| Global Journal of Flexible Systems Management               | Journal            | [3]        | 1   | 2    |
| Real-time Systems                                           | Journal            | [55]       | 1   | 2    |
| VLDB Journal                                                | Journal            | [56]       | 1   | 2    |
| Cluster Computing                                            | Journal            | [57]       | 1   | 2    |
| Journal of Ambient Intelligence and Humanized Computing      | Journal            | [58], [78] | 2   | 3    |
| Journal of Internet Services and Applications               | Journal            | [4]        | 1   | 2    |
| Social Network Analysis and Mining                          | Journal            | [60]       | 1   | 2    |
| International Journal of Parallel Programming               | Journal            | [61]       | 1   | 2    |
| Multimedia Systems                                          | Journal            | [62]       | 1   | 2    |
| International Journal of Intelligent Information and Database Systems | Journal            | [63]       | 1   | 2    |
| International Journal of Information and Decision Sciences   | Journal            | [64]       | 1   | 2    |
| International Journal of Data Analysis Techniques and Strategies | Journal            | [65]       | 1   | 2    |
| International Journal of Grid and High Performance Computing | Journal            | [66]       | 1   | 2    |
| Journal of Database Management                               | Journal            | [67]       | 1   | 2    |
| Journal of Sensors                                          | Journal            | [68]       | 1   | 2    |
| Wireless Communications and Mobile Computing                 | Journal            | [69]       | 1   | 2    |
| The Scientific World Journal                                | Journal            | [70]       | 1   | 2    |
| Future Internet                                             | Journal            | [71]       | 1   | 2    |
| International Journal of Geo-Information                    | Journal            | [72]       | 1   | 2    |
| Cogent Engineering                                          | Journal            | [73]       | 1   | 2    |
| Procedia Environmental Sciences                              | Conference         | [40]       | 1   | 2    |
| IEEE International Conference on Data Engineering           | Conference         | [14]       | 1   | 2    |
| International Conference on Research and Innovation in Information Systems (ICRIIS) | Conference         | [15]       | 1   | 2    |
| IEEE International Conference on Information and Automation | Conference         | [16]       | 1   | 2    |
| IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) | Conference         | [17]       | 1   | 2    |
| International Conference on Digital Information Management (ICDIM) | Conference         | [21], [23], [24] | 3   | 5    |
| IEEE Annual Computer Software and Applications Conference    | Conference         | [22], [28] | 2   | 3    |
| International Seminar on Intelligent Technology and Its Applications (ISITIA) | Conference         | [25]       | 1   | 2    |
| International Conference on Emerging Technologies           | Conference         | [26]       | 1   | 2    |
| IEEE International Conference on Big Data Analysis (ICBDA)   | Conference         | [29]       | 1   | 2    |
Table 7. (Continued.) Publication sources.

| Publication source                                                        | Channel                  | References                  | No. | %|age |
|--------------------------------------------------------------------------|--------------------------|-----------------------------|-----|---|----|
| IEEE International Advance Computing Conference (IACC)                    | Conference               | [30]                        | 1   | 2 |    |
| International Conference on Computer and Information Science             | Conference               | [31]                        | 1   | 2 |    |
| Proceedings of the VLDB Endowment                                       | Conference               | [32], [33]                  | 2   | 3 |    |
| Australasian Database Conference                                        | Conference               | [34]                        | 1   | 2 |    |
| International Conference on Advances in Computing, Communications and Informatics | Conference               | [36]                        | 1   | 2 |    |
| International Conference on Knowledge-Based and Intelligent Information & Engineering Systems | Conference               | [39]                        | 1   | 2 |    |
| International Conference: Beyond Databases, Architectures and Structures | Conference               | [46]                        | 1   | 2 |    |
| British National Conference on Databases                                 | Conference               | [47]                        | 1   | 2 |    |
| Emerging Trends and Applications in Information Communication Technologies | Conference               | [48]                        | 1   | 2 |    |
| Asia-Pacific Web Conference: Web Technologies and Applications          | Conference               | [49]                        | 1   | 2 |    |
| Data Warehousing and Knowledge Discovery                                 | Conference               | [63], [64]                  | 2   | 3 |    |
| International Conference on Future Generation Communication Technologies (FGCT) | Conference               | [71]                        | 1   | 2 |    |
| International Workshop on Real-Time Business Intelligence and Analytics | Workshop                 | [35]                        | 1   | 2 |    |

Real-time stream processing. Key challenges identified in this paper include:

- in-memory computing
- support to semi-structured data streams
- distributed computing
- low latency
- implementation of machine learning algorithms on un-structured big data
- effective resource allocation
- fast disk I/O operation
- distribution of stream engines
- platform independence
- scalability
- real-time processing of spatiotemporal data streams
- fault tolerance
- DBMS migration from SQL to NoSQL
- lock-free concurrent update for moving objects

2) DEVELOPMENTS ADDRESSING IDENTIFIED CHALLENGES PARTICULARLY FOR PROCESSING IoT AND SOCIAL MEDIA DATA STREAMS

Following research activities highlights development of tools/technologies/architectures/frameworks addressing mentioned gaps.

In-memory computing can significantly reduces execution time when input totally fits into memory or multiple iterations over that input required. Experimental analysis of recent real-time processing system (Apache Spark) with Hadoop is presented in [82]. Spark outperforms Hadoop in two experiments when input is on disk and when input is totally cached in RAM due to in-memory processing feature of Spark.

Due to inherent dynamic characteristics of big data its difficult to apply existing data mining tools/technologies. Pre-processing of big data streams, effective resource allocation strategies and parallelization are the issues identified by [6], [38]. Open source tools/technologies for big data analytics such as: Spark streaming, Apache Storm, Splunk stream, Yahoo!S4, NoSQL, Apache Samza etc are highlighted in former study. Big data analysis platforms and tools have been reviewed in a study [3] along with their applications, such as: Hadoop, GridGain, MapReduce, HPCC and Apache Storm. Whereas difficulties in selecting the right stream processing framework were identified and addressed for different use cases while developing a streaming analytics infrastructure by [5]. This study presents critical review of key features of some stream processing engines including Storm, Spark Streaming, Flink, Kafka Streams, IBM Steams. They concluded Kafka Streams and IBM Streams are good options for time-critical application.

A competitive real-time intelligent data processing system name Stream Cube has been implemented in a recent study [54] to handle real-time big data and to bring powerful AI tools into data processing field. Two studies [32], [37] have proposed method and solution for distributed, replicated, and highly available data processing, storage and query system for structured data named: Mesa. Mesa is built using common Google infrastructure and services, including BigTable and Colossus.

To enhance the column store indexes and in-memory tables, [33] proposed solution to significantly improve performance on hybrid workloads. Efficient look ups and column store scan operator also have been addressed in this study. Furthermore, need of approximate computing techniques related to real-time data streams, like: energy-aware approximation, approximation with heterogeneous resources, intelligent data processing and pricing model approximation have been reviewed in another recent study [42].
Moreover, information extraction (IE) techniques required with the rapid growth of multifaceted also called as multi-dimensional unstructured data which are explored in a survey by [51]. Task-dependent and task-independent are the limitations of IE covering all data types. Another study [53] proposed a stream processing framework along with Column Access-aware Instream Data Cache (CAIDC) supporting low response time while maintaining data consistency to migrate RDBMS to NoSQL. Low latency is required while supporting log based trigger in the presence of updates to maintain data consistency and to ensure heavy hitter queries in stream processing framework.

Better resource utilization and real-time scheduling are the key challenges identified and addressed in study [55] for real-time processing of streaming big data. They proposed a hybrid clustering multiprocessor real-time scheduling algorithm and designed real time streaming big data (RT-SBD) processing engine to address these challenges. Their experimental results conclude that proposed solution outperforms the Storm engine in terms of tuple latency, proportional deadline miss ratio, and system throughput.

No dynamic memory allocation, lock free concurrent updates and online pattern detection are the key features required for optimized real-time processing in the applications of large sets of moving objects. These challenges are addressed by the development of multi-layered grid join (MLG-join) algorithm by [56] and a parallel algorithm for timely detection of spatial clusters developed by [58].

\[ a: \text{IoT} \]

To provide real-time services to users in internet of things (IoT) based smart transportation environment, an architecture has been proposed and implemented by [57]. This framework is implemented based on Spark with Hadoop and MapReduce technique that process and handle huge amount of data in real-time. In addition, another IoT based framework developed in a study [60] to analyze students’ performance on real-time basis based on sensor and screen activity data. They applied visual attention techniques for their analysis including: Top-down visual attention, Visual saliency/bottom-up attention, Saliency using natural statistics and A boolean map based saliency. Likewise understanding sensor data and distributed stream engines are the constraints highlighted by [62] and [71].

Processing of geographically distributed data has been surveyed in a study [19], without shifting whole datasets to a single location. In order to address the challenge of scalable processing and low latency for IoT cloud, a robotic application is developed by [72], [73] in another study setup a real time system for processing heterogeneous sensor streams from multiple sources with low latency where Apache Storm is responsible for distributed real time sensor data processing. Authors in [74] proposed a framework to address challenge of continuous growth of massive data streams in a smart city network. This framework consists of three layers, where 2nd layer is responsible for real-time stream processing and data filtering making real-time decisions possible. Proposed framework is then tested with the help of authentic datasets on Hadoop ecosystem proving proposed framework as an improved smart city architecture. Additionally, density based clustering in real-time challenge is addressed by [75] by developing an algorithm which obtains high quality results with low computation time.

A real-time stream processing pipeline and current research activities in real-time spatiotemporal data domain are highlighted and compared by [81] and [80] respectively. Apache Storm, Apache Kafka and GeoMQTT broker are utilized as core tools for the development of pipeline architecture in former study that is capable for real-time processing of spatiotemporal data streams. Whereas, the challenge of event processing capabilities in the area of IoT geospatial architectures is highlighted in latter study. Inconsistency among traditional data access methods and event-driven approaches, and heterogeneous approaches for defining event patterns are few key issues identified by this study which need to be tackled to take full advantage of eventing in GI Science. Esper, Apache Storm, Apache Kafka, ESRI GeoEvent Server and Public Cloud Platforms(Cloud Pub/Sub, AWS IoT Core) are the relevant IoT event processing tools identified in this study.

\[ b: \text{SOCIAL MEDIA} \]

Authors in [76] proposed a method to analyze and process data stream fetched from Twitter data using Hadoop. After analyzing processing time with the use of Hive and Pig on Twitter data, this study conclude Pig appeared more efficient than Hive in terms of execution time and support to semi-structured data. Real-time processing on geolocated data from social media apps using hadoop has been performed in a case study by [77] and implement k-NN model to investigate the power of machine learning algorithms on un-structured big data. Possibility of real-time analysis of huge multimedia stream from online social networks is highlighted in studies [61], [79]. To overcome the difficulty of details consideration of distributed computing and low latency, a framework has been introduced in this study that hides platform details and provide simple interface to programmer. This study provided technical experimental comparison among three big data stream processing applications: Spark Streaming, Storm and Flink. Storm appeared to be slightly faster than Flink whereas Spark performed worst among all during experiments for automatic license plate recognition datasets.

Many researchers have looked into challenges for relational/structured data stream processing and proposed various solutions. Tools and technologies developed for real-time stream processing solutions can be broadly categorized as: Hadoop, Apache Spark, Apache Storm, Splunk Stream, Yahoo!S4, Apache Samza, GridGain, MapReduce, HPCC, Flink, Kafka Streams, IBM Streams, Mesa, Stream Cube, CAIDC, RT-SBD, MLG-join etc. After assessment of selected studies it is found that not much attention has been directed towards unstructured real-time
stream processing. There is a need to put more attention to
the identification of challenges faced during implementation
of unstructured data stream processing for all application
domains. These challenges create opportunities for applica-
tion of new processing technology, which are more suited to
unstructured big data streams.

C. ASSESSMENT OF RQ3: WHICH APPROACHES/TOOLS
HAVE BEEN REPORTED TO ADDRESS CHALLENGES
INTRODUCED AT ETL STAGE WHILE PROCESSING
REAL-TIME STREAM FOR REAL-TIME DWH?
Due to complex and dynamic nature of streaming data,
the analysis of stream processing approaches has become
difficult and challenging. Rigorous studies have been per-
formed for comparative analysis of forty two (42) selected
studies that score in between 3-8 during quality assessment
evaluation addressing real-time stream processing for DWH.
Discussion on requirements/challenges and developments
of approaches addressing identified problems are presented
separately in subsequent subsections:

1) REQUIREMENTS/CHALLENGES FOR REAL-TIME STREAM
PROCESSING FOR REAL-TIME DWH
Following requirements and challenges for implementation
of real-time stream processing for real-time DWH were iden-
tified after exploring various studies [4], [14]–[18], [20]–[31],
[34]–[36], [39]–[41], [43]–[50], [52], [59], [63]–[70], [78]:
• to maintain OLAP availability, recapture consistency
and accuracy, maintain database performance with
changing data sources
• to join distributed stream processing engines and an
external DBMS in order to achieve high performance.
• efficient loading of data streams consisting of complex
events (concatenation of simple events)
• dealing with repeated data streams
• maintaining low memory budget for growing streams
• processing varying attributes of the stream such as data
distribution and arrival rate.
• loading strategy of disk-based relational data blocks
• managing different access rates while joining of growing
streams with disk-based relations
• maintaining regular wait for the join of each stream
• heterogeneous data source integration, data source over-
load, master data overload, schema-less data bases
• continuous availability of databases in case of dis-
tributed DWH
• GUI-based and code-based real-time ETL tools

2) DEVELOPMENTS FOR THE IMPLEMENTATION
OF REAL-TIME DWH
Methods and techniques that have been employed in
analysing real-time streams are outlined in table 8 oldest to
newest order. This comparative analysis has been carried out
based on five main factors: 1) methodology adopted by each
study, 2) challenges identified and addressed in each study,
3) specific tool/data structure used or developed in the design
of each study, 4) supporting shape of data and 5) evidence of
proposed approach.

It is clearly depicted from table 8 that various join algo-
rithms and ETL tools have been developed till today for
improving the efficiency of stream processing (join/queries)
for relational databases/streams addressing challenges iden-
tified during assessment of RQ3. We have classified these
algorithms/approaches into following categories:
• stream-disk join for structured data
• stream-stream join
• sql query decomposition
• multi-join query processing in cloud DWHs
• survey of design approaches from distributed systems,
social media and real-time ETL tools
• architecture/framework for supporting distributed
streaming ETL and data integration in real-time DWH
• development of stream ETL engine
• distributed on demand ETL framework
• code-based real-time ETL tools
Other emerging concept related to near real-time ETL
has been addressed recently in [78]. They identified
and proposed a solution for distributed on demand ETL, and
developed a stream processing framework based on Kafka,
Beam and Spark Streaming. This tool is able to execute work-
loads 10 times faster when compared to other stream pro-
cessing frameworks for near real-time ETL by maintaining
horizontal scalability and fault tolerance. Many researchers
have implemented algorithms based on hash tables/maps
as core data structure and database implementations using
MySQL in their studies as shown in table 8. Identification
of technical implementation details; like methodology and
data structure, will help researchers in further optimization
of existing approaches. However, little attention has been
directed towards implementation of real-time ETL/DWH
models/tools/architectures for structured/semi-structured/un-
structured data streams.

D. ASSESSMENT OF RQ4: WHAT EVIDENCE HAVE BEEN
REPORTED WHILE ADDRESSING DIFFERENT CHALLENGES
FOR PROCESSING REAL-TIME STREAM?
This survey has found out that 51 out of 74 selected studies
contained empirical results. It has been observed that there
has not been any publicly accepted performance benchmark
for real-time stream data processing systems so far, however
we have identified few performance benchmark adopted by
selected studies. Table 8 shows that most of the selected stud-
ies have verified proposed approaches through experimental
evaluation either by making use of synthetic dataset or real-
life dataset or both. Standard benchmark dataset for real-time
streaming analytics has not been widely adopted. However,
few of the researchers that used standardized benchmarking
are briefly discussed below.

Authors in [16], [22], [28] validated their approaches by
making use of TPC-H benchmark whereas, new TPC-DS
benchmark has been used in experiments by [31]. The data
| Algorithm/Approach | Methodology | Challenges Addressed | Ref. | Tool/Data Structure | Shape of Data | Evidence |
|-------------------|-------------|----------------------|------|---------------------|---------------|----------|
| Similarity Join Processing on Uncertain Data Streams (USJ) & adaptive superset pre-join (ASP) | framework using sliding window concept | effective pruning methods on both object and sample levels to filter out false alarms, efficient query procedure that can incrementally maintain the USJ answers and to maintain a superset of USJ candidate pairs | [17] | USJ framework | Objects | Performance of USJ is tested using two real data sets and complexities are analyzed using proposed cost model |
| Solution for 24/7 real-time data loading into DWH (RTDW) | every DWH database schema replicated and updated simultaneously, but only one is available for OLAP applications | a simple, fast and efficient solution enabling continuous data loading and OLAP availability on a 24/7 schedule | [28] | 24/7 RTDW architecture, MySQL | Relational | Experimental evaluations using real-world DW and the TPC-H decision support benchmark |
| Semi-Streaming Index Join (SSIJ) framework | index-based stream-disk join algorithm for joining a relational stream with a disk resident relation | amortization of expensive disk seeks for blocks of the stored relation among a large number of stream tuples | [14] | B-tree, hash tables, bitmapped indexes | Relational | Experimetted on synthetic and real-life data sets using cost model |
| A burst resolution technique for data streams management | operational data store is used as staging area in proposed framework to store extracted files. This framework process incoming continuous data that can not be handled by ETL tools due to their special characteristics | to minimize drop of data streams, synchronize processing technology with them, resolve streaming bursts and load balancing | [27] | token bucket technique | Relational | Experimentally validated based on synthetic dataset |
| Extended Hybrid Join (X-HYBRIDJOIN) | stream-disk join algorithm that uses two-level hash table for the joining of fast incoming stream tuples by using a partition based waiting area to store other stream tuples | can adapt to data skew and stores parts of the master data in memory permanently, reducing the disk access overhead | [45], [47] | hash table, linked list, MySQL | Relational | Experimented on synthetic data sets using proposed cost model |
| Tuned "X-HYBRIDJOIN" | stream-disk join tuning module for existing X-HYBRIDJOIN | optimization of memory distribution to the swappable parts of disk buffer | [36], [48], [49] | hash table, MySQL | Relational | Experimented on synthetic data sets using revised cost model |
| 24/7 availability and performance of distributed real-time DWH | standard DW-Striping (DW-S) round-robin technique is used to distribute portions of each fact table among pairs of slave nodes, which is an exact replica of its partner, allows in balancing query execution and replacing any defective node | Solution for continuous availability for distributed DWBS with frequent data loading requirements | [22] | DW-S technique, MySQL | Relational | experiments using the TPC-H decision support benchmark to evaluate the scalability of the proposed solution |
| Optimised X-HYBRIDJOIN | stream-disk join algorithm with two phases: one dealing with frequently accessed disk-relation data while other dealing with other part | improving performance by treating frequently accessed data differently | [34] | hash table, linked list, MySQL | Relational | Validation of cost model using synthetic data sets |
| Scheduling strategies for efficient ETL execution | four scheduling policies on different flow structures and configurations are investigated experimentally. Three generic algorithms were explored in this study: ROUND ROBIN, MINIMUM COST PREDICTION, and MINIMUM MEMORY PREDICTION and an extensible solution for implementing an ETL scheduler is proposed | ETL optimization in terms memory consumption and execution time | [41] | ETL structural workflow patterns, MySQL | Relational | Experimentally validated using synthetic data sets |
| Algorithm/Approach | Methodology | Challenges Addressed | Ref. | Tool/Data Structure | Shape of Data | Evidence |
|--------------------|-------------|----------------------|------|---------------------|---------------|----------|
| Semi-Stream Join (SSJ), Semi-Stream Balanced Join (SSBJ) algorithms | stream disk join algorithm with a cache module, that alternates between stream-probing and disk-probing phases, and many to many semi-stream join | Reducing I/O cost by storing the frequently occurring items of stream in memory | [21], [26], [50] | hash linked tables, list, MySQL | Relational | validated using synthetic and real-life data sets |
| SSCJ: semi-stream cache join algorithm | A front staged cache module is used which takes stream as input and other module uses disk relation as input | optimize performance regarding memory and processing cost | [64] | linked list queue, hash table, MySQL | Relational | validated using synthetic and real-life data sets |
| Optimizing queue-based semi-stream joins | a strategy is proposed which takes turn between last and first element of queue for lookup, along with basic modules designed by same authors | for high probability partitions, this methodology reduces the saturation effect and frequently hitting high-probability | [63] | queue and hash table | Relational | Experimentally validated using real-life and synthetic data sets |
| Development of stream ETL engine (StrETL) | stream data sets stored in a Stream Materialized Aggregate List (StrDW[MAL]) | creation of an interface between StrETL-RT and StrMAL engines and load balancing | [46] | queue, MySQL, StrETL-RT engine | Flat and spatial data, sensor data | Experimentally validated |
| Two-Level data Partitioning Approach for Real-Time Data Warehouse (2LPA-RTDW) | this approach finds the right number of clusters and divides DWH into partitions equally using the horizontal partitioning (Greenspan based fragmentation approach), in the second-level data partitioning, 2LPA-RTDW try to keep the balance of data amount in each partition by merging and dividing the existing ones | high refreshing frequency | [31] | Matrix, 2LPA-RTDW | Relational | proposed approach is evaluated using the new TPC-DS benchmark |
| Cached-based stream-disk join algorithm (CSDJ) along with optimization | two complementary hash join phases: disk-probing phase and stream-probing phase | exploits skew characteristic in stream data more appropriately | [20], [23], [69], [70] | hash table, linked list, MySQL | Relational | Experimentally evaluated using synthetic and real-life data sets |
| Real-Time Data ETL Framework for Big Real-Time Data Analysis | stream-disk join: a dynamic mirror replication technology was proposed to avoid the contention between OLAP queries and OLTP update, using external dynamic storage area(DSA) | query contention and data skew | [16] | linked list, CDC, DSA | Relational | Benchmark TPC-H is adopted to evaluate the pre-processing performance |
| ETL modeling with the simplification, adjustment, and design of ETL scenarios | a model that aims to design ETL scenarios, customize and simplify the mapping between the attributes in the data source with the attributes of the data warehouse tables | automatic data preprocessing that regulates the insertion of new data and update data without generating a lot of queries | [40] | vector geometry | Relational | Tested with real-life data sets |
| Skewed Distributions in Semi-Stream Joins | two optimization techniques for frequently used master data and for selective load shedding for stream tuples | to improve service rate for typical data with skewed distribution | [43] | hash tables, linked lists | Relational | Experimentally evaluated using synthetic and real-life data sets |
| Algorithm/Approach                                      | Methodology                                                   | Challenges Addressed                                                                 | Ref. | Tool/Data Structure                           | Shape of Data       | Evidence                                      |
|-------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------|------|-----------------------------------------------|---------------------|-----------------------------------------------|
| Comparative review of DWH ETL tools                   | comparative review based on near real-time ETL approaches like: Change Data Capture (CDC), Trickle and Flip approach, real time data cache (RTDC) approach | market value and relevance of ETL tools in data science industry, and growing need of real time data analysis from structured and unstructured data sources | [30] | ETL tools (Informatica, Datastage, Ab Initio, Oracle Data Integrator, SSIS) | Relational, PDF, XML etc | Industry insights for the relevance of tools |
| Survey of design approaches for DWH from social media | literature review                                              | exploitation of data from the web                                                   | [4]  | Multidimensional model for DWH (Conceptual, Logical) | heterogeneous social media data | No evidence provided in this literature review |
| Identification of challenges of ETL implementation for near real-time environment | literature review to find challenges and solution approaches for ETL implementation | high availability, low latency and horizontal scalability features for functionality | [15], [25] | Change data capture (CDC), CDC log-based, real-time data cache (RTDC), Trickle and Flip | Relational | No evidence provided in this literature review |
| Optimized foreign-key join algorithm for OLAP workloads | foreign-key join algorithm instead of general-purpose hash joins | to enable surrogate key index to be efficient for foreign key joins in DWH workloads for both hardware accelerators and CPU | [59] | array-store oriented foreign key, Xeon Phi and NVIDIA A100 GPU platforms | Relational | Proposed approach is evaluated using synthetic and real-life datasets |
| Performance Analysis of Not Only SQL Semi-Stream Join Using MongoDB | join module of stream with disk based data                    | efficient stream processing for NoSQL data for real-time DWH                        | [24] | MongoDB | NoSQL, Relational | Experimentally evaluated using synthetic and real-life data sets |
| Framework for big DWH dealing in both real-time and offline modes | first component: real-time data ingestion of both streaming and offline data generated by communication service providers (CSP) and coding, second component: big data ETL module | to effectively organize raw data, and implement complex and more intelligent use-cases that help in improving core networks and other areas of CSP | [29] | NiFi, Kafka, Spark Streaming, Hadoop | Relational | No experimental evidence provided |
| Architecture of Striim’s streaming ETL engine, a distributed streaming ETL and intelligence platform | to handle the demands of modern data pipelines, transformation engine has been designed. Open Processor component of this engine enables users to develop join functionality or run machine learning models on the input data streams | to run low-latency transformation logic on input data streams using modern approaches of query optimization and execution. To enable declarative data filtering and updation on streaming real-time data | [35] | Striim transformation engine, SQL, CDC, Kafka, open source streaming ETL engine KSQL | Relational | Experimentally evaluated based on synthetic and real-life datasets |
| Overview of the existing data quality approaches in the ETL process | comparative review of some commercial ETL tools considering highlighted data quality characteristics (three quality dimensions considered for comparison: performance, reliability, deduplication): ETL tools: Talend Data Quality (TDQ), Talend Data Integration (TDI), Pentaho Data Integration (PDI), Informatica Data Integration (IDI) and Microsoft SQL Server Integration Services (SSIS) | management of data from internal and external sources: data with heterogeneous content and diverse quality problems | [39] | ETL tools: TDQ, TDI, PDI, IDI, SSIS | Relational | Comparative evaluation based on TDQ and TDI tools |
| Algorithm/Approach | Methodology | Challenges Addressed | Ref. | Tool/Data Structure | Shape of Data | Evidence |
|-------------------|-------------|----------------------|------|---------------------|---------------|----------|
| **Comparative evaluation of code-based ETL tools and modeling of a real-time ETL process with incremental loading** | three parts of model: snapshot-based CDC, dimension processing algorithm and finally an algorithm for fact processing | to reduce data flow and latency using code-based real-time ETL tools | [52] | Code-based ETL tools: Pygramed, Peti, Scriptella, R-etl | Relational, CSV | Experimentally evaluated based on synthetic dataset |
| **A feedback control scheduling architecture for real-time DWH (FCSA-RTDW) and two-level data partitioning approach (2LPA-RTDW)** | an architecture called DETL-(m, k)-firm-real-time DWH architecture (decentralised extract-transform-load approach based on (m, k)-firm constraints for real-time DWH) | increase in query processing speed and reducing those transactions which don’t meet their deadline | [65], [66] | FCSA-RTDW architecture, 2LPA-RTDW, MySQL | Relational | Experimentally evaluated based on simulation results |
| **The conceptual modeling, the architecture and loading methodology of the real-time DWH (RTDW)** | three parts: - loading data into cache levels - loading data into real-time storage area - loading data into the static DWH | to handle complex/multiple event streams such as: spatial, semantic, temporal and real-time | [67] | RTDW architecture, XML data format | Various DB systems | Experimentally evaluated based on real-life dataset |
| **Optimizing Communication for Multi-Join Query Processing in Cloud DWIs** | PK-map and Tuple-index-map are the proposed data structures | to allocate resource dynamically on demand for growing data size in cloud environment during frequent data integration into DWH | [68] | PK-map, tuple-index-map as storage structures | Unstructured dataset | Experimentally evaluated |
| **Extension of Spark Streaming, DSjoin for distributed processing** | implementation of DS-join, stream-disk join operator to process hit and missed data in parallel | load balancing among multiple databases, parallelizing the join processing, reducing number of database accesses, and managing cache and data shuffling | [18] | Apache Spark | Relational, NoSQL | Experimentally evaluated based on synthetic and real-life datasets |
| **Optimization of near real-time ETL based Kafka, Beam and Spark Streaming** | it uses in-memory master data cache to optimize performance, a buffer for join operations on data with different arrival rate to synchronize consistency, and a unified programming model to allow this tool to be used on top of a number of stream processing frameworks | fulfills limitations of high availability, low latency and horizontal scalability by on-demand data stream, steps performed in distributed and parallel manner, in-memory cache, un-synchronized consistency, unified programming model | [78] | Kafka, H2, Spark | Beam, Apache Spark | Relational | Experimentally evaluated based on synthetic and real-life data sets |
| **A rewrite/merge framework for supporting real-time data warehousing via lightweight data integration** | a novel DWH framework separating static phase from dynamic phase to obtain real-time processing features | fulfills limitations of actual DWH architectures, which are not appropriate to execute classical operations under real-time constraints such as: loading, indexing, aggregation, OLAP query answering etc. | [44] | B-tree, Bitmap indexes, CDC, MySQL | Relational | Experimentally evaluated based on synthetic data |
was collected using a single Intel Haswell CPU with 12 cores, 2.30 GHz and hyper-threading disabled to analyse query performance by [33]. Whereas Yahoo! Cloud Serving Benchmark has been used to test the performance of work in study [53]. Some real-world trajectory datasets have been adopted by [56] and [58]: a fleet of trucks, a city buses for experimental evaluation of proposed methodologies whereas, synthetic datasets generated using the benchmark data generator were also used during evaluation by [56]. Semi-stream join algorithms developed by [20], [23], [43], [69], [70] were tested by using both synthetic and real-life datasets. They also analyzed memory and time requirements. In addition, a modified well-known Star Schema Benchmark (SSB), called SSB-RT, is used during experimental evaluation of rewrite/merge framework by [44], which embeds real-time features, as well as TPC-H benchmark.

While validating optimized algorithms addressing common challenges for IoT stream processing, one of the selected studies [75] use KDD CUP99 Network Intrusion Detection dataset that comes from the 1998 DARPA Intrusion Detection as real-life data set along with 3 synthetic datasets.

In addition, to strengthen the evidence of their approaches, various studies have been identified in this literature which derived cost models to compute data latency and resources utilization: [14], [17], [20], [23], [36], [43], [45], [47], [55], [69], [70]. Moreover, research efforts should be directed towards advancing appropriate benchmarks for evaluating different real-time stream processing applications. This would go a long way to minimize data latency and resources utilization.

V. CONCLUDED DISCUSSION

The objective of this survey is to provide guidance for researchers in the subject of real-time stream analysis for DWH and big data applications. For this purpose, we have investigated applications, developments and challenges in terms of methodology, data structure used and shape of data. Our exploration highlights main target publication channels for real-time stream processing research in real-time DWH domain and in other big data applications such as: (IoT, Social media, Google etc). Our literature further highlights implementation challenges along with developed approaches/tools and evaluation evidences for real-time stream processing in all mentioned application domains.

A. GENERAL OBSERVATIONS

Observations from the literature reveal that there exists various algorithms for implementing real-time join processing at ETL stage for structured data and there seem to be less tools and technologies that offer real-time join processing for unstructured data. It is further observed that little attention of researchers is found in existing studies to discuss the data structures used by every approach for the development of their algorithms which if exists might help researchers to address research gaps in this subject domain. Research efforts should be geared towards advancing processing approaches that are suitable for all existing/important type of streaming data. In addition, it is rare to find specific algorithm/approach that collectively addressed all identified challenges. It is further observed that many researchers have looked into different features and suitability of existing stream processing engines, however there is still need for the development of stream engines flexible for modification according to business needs.

B. FUTURE DIRECTIONS

Fitting DWH in cloud architecture is of tremendous business need as cloud storage is economical and scalable. If DWH is integrated with cloud computing it can handle relational and non-relational data and can be offered as a service. More tools and technologies could be developed for implementing cloud DWH concept. Moreover, purpose build ETL tools that are ready to connect open data sources or which can delay transformation phase until it is needed are becoming more effective from industry’s viewpoint.

REFERENCES

[1] N. Sharma, A. Iyer, R. Bhattacharya, N. Modi and W. Crivelini, Getting Started With Data Warehousing, 1st ed. Markham, ON, Canada: IBM Canada, 2012.
[2] P. Ponniah, Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals, Hoboken, NJ, USA: Wiley, 2004.
[3] P. Grover and A. K. Kar, “Big data analytics: A review on theoretical contributions and tools used in literature,” Global J. Flexible Syst. Manage., vol. 18, no. 3, pp. 203–229, Sep. 2017.
[4] I. Moulla, A. Nabi, L. Bouguenda, and M. Hammami, “Data warehouse design approaches from social media: Review and comparison,” Social Netw. Anal. Mining, vol. 7, no. 1, p. 5, Dec. 2017.
[5] H. Isah, T. Abughofa, S. Mahfuz, D. Ajeria, F. Zulkernine, and S. Khan, “A survey of distributed data stream processing frameworks,” IEEE Access, vol. 7, pp. 154300–154316, 2019.
[6] T. Kolajo, O. Daromudana and A. Adebiyi, “Big data stream analysis: A systematic literature review,” J. Big Data, vol. 6, no. 1, p. 47, Dec. 2019.
[7] P. Breton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” J. Syst. Softw., vol. 80, no. 4, pp. 571–583, Apr. 2007.
[8] B. Kitchenham, “Procedures for performing systematic reviews,” Keele Univ., Keele, U.K., Tech. Rep. TR/SE-0401, 2004, vol. 33, pp. 1–26.
[9] A. Fernandez, E. Insfran, and S. Abrabaho, “Usability evaluation methods for the Web: A systematic mapping study,” Inf. Softw. Technol., vol. 53, no. 8, pp. 789–817, Aug. 2011.
[10] S. Oubbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, “Requirements engineering education: A systematic mapping study,” Requirements Eng., vol. 20, no. 2, pp. 119–138, Jun. 2015.
[11] (2018). CORE Conference Portal. Accessed: Jan. 30, 2020. [Online]. Available: http://portal.core.edu.au/conf-ranks/
[12] (2018). Scimago Journal Country Rank. Accessed: Jan. 30, 2020. [Online]. Available: https://www.scimagojr.com/
[13] C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” in Proc. 18th Int. Conf. Eval. Assessment Softw. Eng. (EASE), 2014, p. 38.
[14] M. A. Horna, A. Deligiannakis, Y. Kotidis, and V. Vassalos, “Semi-structured index join for near-real time execution of ETL transformations,” in Proc. IEEE 27th Int. Conf. Data Eng., Apr. 2011, pp. 159–170.
[15] A. Sabtu, N. F. M. Azmi, N. N. A. Sjarif, S. A. Ismail, O. M. Yusop, H. Sarkan, and S. Chuprat, “The challenges of extract, transform and loading (ETL) system implementation for near real-time environment,” in Proc. Int. Conf. Res. Innov. Inf. Syst. (ICRIS), Jul. 2017, pp. 1–5.
[16] X. Li and Y. Mao, “Real-time data ETL framework for big real-time data analysis,” in Proc. IEEE Int. Conf. Automat., Aug. 2015, pp. 1289–1294.
[17] X. Li and L. Chen, “Similarity join processing on uncertain data streams,” IEEE Trans. Knowl. Data Eng., vol. 23, no. 11, pp. 1718–1734, Nov. 2011.

[18] Y.-H. Jeon, K.-H. Lee, and H.-J. Kim, “Distributed join processing between streaming and stored big data under the micro-batch model,” IEEE Access, vol. 7, pp. 6,383–3,498, 2019.

[19] S. Dolev, P. Florissi, E. Gudes, S. Sharma, and I. Singer, “A survey on geographically distributed big-data processing using MapReduce,” IEEE Trans. Big Data, vol. 5, no. 1, pp. 60–80, Mar. 2019.

[20] E. Mehmood and M. A. Naeem, “Optimization of cache-based semi-stream joins,” in Proc. IEEE 2nd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA), Apr. 2017, pp. 76–81.

[21] M. A. Naeem, “Efficient processing of semi-stream data,” in Proc. 8th Int. Conf. Digit. Inf. Manage. (ICDIM), Sep. 2013, pp. 7–10.

[22] R. J. Santos, J. Bernardino, and M. Vieira, “24/7 real-time data warehousing,” in Proc. IEEE 36th Annu. Comput. Softw. Appl. Conf. (JICDM), Jul. 2012, pp. 654–659.

[23] M. A. Naeem, I. S. Bajwa, and N. Jamil, “A cached-based approach to enrich stream data with master data,” in Proc. 10th Int. Conf. Digit. Inf. Manage. (ICDIM), Oct. 2015, pp. 57–62.

[24] E. Mehmood and T. Anees, “Performance analysis of not only SQL semi-stream join using MongoDB for real-time data warehousing,” IEEE Access, vol. 7, pp. 13,421–13,4225, 2019.

[25] A. Wibowo, “Problems and available solutions on the stage of extract, transform, and loading in near-real time data warehousing (a literature study),” in Proc. Int. Seminar Technol. Appl. (ISITA), 2015, pp. 345–350.

[26] M. A. Naeem, “A robust join operator to process streaming data in near-real-time,” in Proc. Int. Conf. Digit. Inf. Manage. (ICDIM), Sep. 2013, pp. 119–124.

[27] F. Majeed, S. Mahmood, S. Ubaid, N. Khalil, S. Siddiqui, and F. Ashraf, “A burst resolution technique for data streams management in the real-time data warehouse,” in Proc. 7th Int. Conf. Emerg. Technol., Sep. 2011, pp. 1–5.

[28] R. J. Santos, J. Bernardino, and M. Vieira, “24/7 real-time data warehousing: A tool for continuous actionable knowledge,” in Proc. IEEE 35th Annu. Comput. Softw. Appl. Conf. (JICDM), Jul. 2011, pp. 279–288.

[29] A. R. Ali, “Real-time big data warehousing and analysis framework,” in Proc. IEEE 8th Int. Conf. Big Data Anal. (ICBDA), Mar. 2018, pp. 43–49.

[30] R. Mukherjee and P. Kar, “A comparative review of data warehousing ETL tools with new trends and industry insight,” in Proc. IEEE 7th Int. Advance Comput. Conf. (IACC), Jan. 2017, pp. 943–948.

[31] I. Hamdi, E. Bouazizi, S. Alshomrani, and J. Feki, “2LPA-RTDW: A two-level data partitioning approach for real-time data warehouse,” in Proc. IEEE/ACIS 14th Int. Conf. Comput. Inf. Sci. (ICIS), Jun. 2015, pp. 632–638.

[32] A. Gupta, F. Yang, J. Govig, A. Kirsch, K. Chan, K. Lai, S. Wu, S. G. Dho, A. R. Kumar, A. Agiwal, S. Bhansali, M. Hong, J. Cameron, M. Siddiqui, D. Jones, J. Shute, A. Gubarev, S. Venkataraman, and D. Agrawal, “Mesa: Geo-replicated, near-real-time, scalable data warehousing,” Proc. VLDB Endowment, vol. 7, no. 12, pp. 1259–1270, Aug. 2014.

[33] P.-A. Larson, A. Birka, E. N. Hanson, W. Huang, M. Nowakiewicz, and V. Papadimos, “Real-time analytical processing with SQL server,” Proc. VLDB Endowment, vol. 8, no. 12, pp. 1740–1751, Aug. 2015.

[34] M. A. Naeem, G. Dobbie, and G. Weber, “Optimised X-HYBRIDJOIN for near-real-time data warehousing,” in Proc. 23rd Australas. Database Conf., vol. 24, no. 4, pp. 391–406, Jul. 2018.

[35] M. Babar and F. Arif, “Real-time data processing scheme using big memory resources in near-real-time data warehousing,” Proc. VLDB Endowment, vol. 11, no. 11, pp. 623–649, Dec. 2018.

[36] K. Adnan and R. Akbar, “An analytical study of information extraction from unstructured and multidimensional big data,” J. Big Data, vol. 6, no. 1, p. 91, Dec. 2019.

[37] N. Biswas, A. Sarkar, and K. C. Mondal, “Efficient incremental loading in ETL processing for real-time data integration,” Innov. Syst. Softw. Eng., vol. 16, pp. 203–221, May 2019.

[38] K. Ma and B. Yang, “Column access-aware in-stream data cache with streaming framework processing,” J. Signal Process. Syst., vol. 86, nos. 2–3, pp. 191–205, Mar. 2017.

[39] T. Zheng, G. Chen, X. Wang, C. Chen, X. Wang, and S. Luo, “Real-time intelligent big data processing: Technology, platform, and applications,” Sci. China Inf. Sci., vol. 62, no. 8, p. 82101, Aug. 2019.

[40] A. A. Safaei, “Real-time processing of streaming big data,” Real-Time Syst., vol. 53, no. 1, pp. 1–44, Jan. 2017.

[41] P. G. D. Ward, Z. He, R. Zhang, and J. Qi, “Real-time continuous intersecion joins over large sets of moving objects using graph processing units,” VLDB J., vol. 23, no. 6, pp. 965–985, Dec. 2014.

[42] M. Babar and F. Arif, “Real-time data processing scheme using big data analytics in Internet of Things based smart transportation environment,” J. Ambient Intell. Hum. Comput., vol. 10, no. 10, pp. 4167–4177, Oct. 2019.

[43] M. R. Junior, B. Olivier, and M. Endler, “DG2CEP: A near real-time on-line algorithm for detecting spatial clusters large data streams through complex event processing,” J. Internet Services Appl., vol. 10, no. 1, p. 8, Dec. 2019.

[44] Y. Zhang, Y. Zhang, X. Zhou, and J. Lu, “Main-memory foreign key joins on advanced processors: Design and re-evaluations for OLAP workloads,” Distrib. Parallel Databases, vol. 37, no. 4, pp. 469–506, Dec. 2019.

[45] M. Farhan, S. Jabbar, M. Aslam, A. Ahmad, M. M. Iqbal, M. Khan, and M.-E. A. Maria, “A real-time data mining approach for interaction analytics assessment: IoT-based student interaction framework,” Int. J. Parallel Program., vol. 46, no. 5, pp. 886–903, Oct. 2018.

[46] I. Bartolini and M. Patella, “A general framework for real-time analysis of massive multimedia streams,” Multimedia Syst., vol. 24, no. 4, pp. 391–406, Jul. 2018.
62. L. Rodríguez-Mazahua, C.-A. Rodríguez-Enríquez, J. L. Sánchez-Cervantes, J. Cervantes, J. L. García-Alcaraz, and G. Alor-Hernández, “A general perspective of big data: Applications, tools, challenges and trends,” J. Supercomput., vol. 72, no. 8, pp. 3073–3113, Aug. 2016.

63. M. A. Naeem, G. Weber, G. Dobbie, and C. Lutteroth, “SSCJ: A semi-stream cache join using a front-stage cache module,” in Proc. Int. Conf. Data Warehousing Knowl. Discovery, Cham, Switzerland: Springer, 2014, pp. 171–182.

64. M. A. Naeem, G. Weber, G. Dobbie, and C. Lutteroth, “SSCJ: A semi-stream cache join using a front-stage cache module,” in Proc. Int. Conf. Data Warehousing Knowl. Discovery, Cham, Switzerland: Springer, 2014, pp. 236–247.

65. I. Hamdi, E. Bouazizi, and J. Feki, “Query optimisation in real-time data warehouses,” Int. J. Intell. Inf. Database Syst., vol. 12, no. 4, pp. 245–278, 2019.

66. I. Hamdi, E. Bouazizi, S. Alshomrani, and J. Feki, “Improving QoS in real-time data warehouses by using feedback control scheduling,” Int. J. Inf. Decis. Sci., vol. 10, no. 3, pp. 181–211, 2018.

67. H. Bouali, J. Akaichi, and A. Gaaloul, “Real-time data warehouse loading methodology and architecture: A healthcare use case,” Int. J. Data Anal. Techn. Strategies, vol. 11, no. 4, pp. 310–327, 2019.

68. S. Kurunji, T. Ge, X. Fu, B. Liu, and C. X. Chen, “Optimizing communication for multi-join query processing in cloud data warehouses,” Int. J. Grid High Perform. Comput., vol. 5, no. 4, pp. 113–130, Oct. 2013.

69. M. A. Naeem, E. Mehmood, M. G. A. Malik, and N. Jamil, “Optimizing semi-stream CACHEJOIN for near-real-time data warehousing,” J. Database Manage., vol. 31, no. 1, pp. 20–37, Jan. 2020.

70. M. A. Naeem, I. S. Bajwa, and N. Jamil, “A cached-based stream-relation join operator for semi-stream data processing,” Int. J. Data Warehousing Mining, vol. 12, no. 3, pp. 14–31, Jul. 2016.

71. H. S. Jung, C. S. Yoon, Y. W. Lee, J. W. Park, and C. H. Yun, “Cloud computing platform based real-time processing for stream reasoning,” in Proc. 6th Int. Conf. Future Gener. Commun. Technol. (FGCT), Aug. 2017, pp. 1–5.

72. S. Kamburugamuve, L. Christiansen, and G. Fox, “A framework for real time processing of sensor data in the cloud,” J. Sensors, vol. 2015, pp. 1–11, Apr. 2015.

73. J. Hu, R. Sun, F. Wang, X. Fei, and K. Zhao, “A stream processing system for multisource heterogeneous sensor data,” J. Sensors, vol. 2016, pp. 1–8, May 2016.

74. B. N. Silva, M. Khan, and K. Han, “Big data analytics embedded smart city architecture for performance enhancement through real-time data processing and decision-making,” Wireless Commun. Mobile Comput., vol. 2017, pp. 1–12, Jan. 2017.

75. A. Amini, H. Saboohi, T. Y. Wah, and T. Herawan, “A fast density-based clustering algorithm for real-time Internet of Things stream,” Sci. World J., vol. 2014, pp. 1–11, Jun. 2014.

76. A. P. Rodrigues and N. N. Chipulkaran, “Real-time Twitter data analysis using Hadoop ecosystem,” Cogent Eng., vol. 5, no. 1, Oct. 2018, Art. no. 1534519.

77. J. Dicomas, “Processing unstructured documents and social media using big data techniques,” Econ. Res.-Ekonomiska Istraživanja, vol. 28, no. 1, pp. 981–993, Jan. 2015.

78. G. V. Machado, I. Cunha, A. C. M. Pereira, and L. B. Oliveira, “DOD-ETL: Distributed on-demand ETL for near real-time business intelligence,” J. Internet Services Appl., vol. 10, no. 1, p. 21, Dec. 2019.

79. I. Bartolini and M. Patella, “Real-time stream processing in social networks with RAM3S,” Future Internet, vol. 11, no. 12, p. 249, Nov. 2019.

80. R. M. Rieke, L. Bigagli, S. Herle, S. Jirka, A. Koste, T. Liebig, C. Malewski, T. Paschke, and C. Stasch, “Geospatial IoT—The need for event-driven architectures in contemporary spatial data infrastructures,” ISPRS Int. J. Geo-Inf., vol. 7, no. 10, p. 385, 2018.

81. M. Laska, S. Herle, R. Klamma, and J. Blankenbach, “A scalable architecture for real-time stream processing of spatiotemporal IoT stream data—Performance analysis on the example of map matching,” ISPRS Int. J. Geo-Inf., vol. 7, no. 7, p. 238, 2018.

82. S. Shahrivari, “Beyond batch processing: Towards real-time and streaming big data,” Computers, vol. 3, no. 4, pp. 117–129, Oct. 2014.

83. H. Han, H. Jung, H. Eom, and H. Y. Yeom, “Scatter-gather-merge: An efficient star-join query processing algorithm for data-parallel frameworks,” Cluster Comput., vol. 14, no. 2, pp. 183–197, Jun. 2011.

ERUM MEHMOOD was born in Pakistan. She received the M.Phil. degree in computer science from NCBAE, Lahore, Pakistan, in 2017. Her M.Phil. dissertation is in the area of stream processing for real-time data warehousing. She is currently pursuing the Ph.D. degree with the University of Management and Technology, Lahore, under the supervision of Dr. Tayyaba Anees. She is currently working as a Lecturer of computer science with the Government Degree College, Lahore. Her research interests include big data analytics, stream processing, ETL, and real-time data warehousing.

TAYYABA ANEES was born in Pakistan. She received the Ph.D. degree from the Vienna University of Technology, Vienna, Austria, in 2012. Her Ph.D. dissertation is in the area of service-oriented architecture and web services availability domain. She has worked as the Project Assistant at the Vienna University of Technology for four years. She is currently working as the Director Software Engineering Program/Assistant Professor at the Software Engineering Department, University of Management and Technology, Lahore. Her research interests include service-oriented architecture, web services, software availability, software safety, software engineering, software fault tolerance, and real-time data warehousing.

**E. Mehmoood, T. Anees: Challenges and Solutions for Processing Real-Time Big Data Stream**