# Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges

**Conference Paper** · October 2018

DOI: 10.1109/ISMSIT.2018.8567061



**2 authors:**

Fatih Gurcan
Karadeniz Technical University
**14** PUBLICATIONS **8** CITATIONS

SEE PROFILE

Muhammet Berigel
Karadeniz Technical University
**15** PUBLICATIONS **20** CITATIONS

SEE PROFILE

# Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges

Fatih Gürcan
*Department of Computer Engineering*
*Karadeniz Technical University*
Trabzon, Turkey
fgurcan@ktu.edu.tr

Muhammet Berigel
*Department of Management Information Systems*
*Karadeniz Technical University*
Trabzon, Turkey
berigel@ktu.edu.tr

***Abstract*—In today's technological environments, the vast majority of big data-driven applications and solutions are based on real-time processing of streaming data. The real-time processing and analytics of big data streams play a crucial role in the development of big-data driven applications and solutions. From this perspective, this paper defines a lifecycle for the real-time big data processing. It describes existing tools, tasks, and frameworks by associating them with the phases of the lifecycle, which include data ingestion, data storage, stream processing, analytical data store, and analysis and reporting. The paper also investigates the real-time big data processing tools consisting of Flume, Kafka, Nifi, Storm, Spark Streaming, S4, Flink, Samza, Hbase, Hive, Cassandra, Splunk, and Sap Hana. As well as, it discusses the up-to-date challenges of the real-time big data processing such as "volume, variety and heterogeneity", "data capture and storage", "inconsistency and incompleteness", "scalability", "real-time processing", "data visualization", "skill requirements", and "privacy and security". This paper may provide valuable insights into the understanding of the lifecycle, related tools and tasks, and challenges of real-time big data processing.**

***Keywords—Big data streams, real-time big data processing, lifecycle, tools, tasks, challenges.***

## I. INTRODUCTION

In line with the developments formed under the leadership of information and communication technologies (ICTs), big data has become one of the most rapidly growing trends in recent years. From this perspective, big data is considered by many authorities as tomorrow's data architecture, and also it is regarded as one of the top 10 crucial technologies that will change the world [1]. In a general sense, "big data" is defined by the three fundamental features including volume, variety, velocity [2]. These dimensions describe the landscape of big data. Through the efficient use of many online resources such as internet of things (IoT), mobile devices, social networks, and sensors, the number of real-time applications of big data streams has increased considerably. The introduction of big data technologies has led to a great transformation in the methodologies of data progressing and analytics. In today's competitive business environments, the progressing and analytics of big data plays an important role in achieving successful business strategies [1], [3]. Big data processing is becoming a reality in many real-world applications and solutions [1], [4]–[6].

In general sense, the big data processing can be categorized into three distinct types, comprising batch processing, real-time processing, and hybrid (batch and real-time) processing. Batch data processing is a well-organized technique of processing high volumes of data. The transactions in this paradigm are carried out within a specific period of time. Hadoop is the most common framework used for batch processing. Batch processing involves distinct transactions for ingestion, processing, analytics, and reporting [1], [4]–[6]. In this paradigm, data are ingested, stored into the databases, and processed. The batch outcomes are analyzed and then produced. On the other hand, numerous big data applications and solutions require real-time processing of big data (streams). Real-time processing consists of continuous input, processing, and analysis and output (reporting) of data. This processing paradigm aims the lowest-latency during the process. In this context, many frameworks are available for real-time big data processing like Storm, Spark, S4, Flink, Samza [1], [3]–[6].

Compared to real-time processing, batch processing is not time-limited, and so it is possible to perform more comprehensive analysis and achieve more effective results. But this paradigm is not suitable for applications and solutions that require a low response time. In some cases, real-time applications may need to execute with a low response time. In such cases, the processing and analysis of data is commonly restricted in order to obtain low response time. To overcome this difficulty, hybrid processing approach has been suggested as a new paradigm. The hybrid processing approach is required for many big data application and solution domains that employ batch as well as real-time processing. The results obtained by using batch and real-time processing together are analyzed and queried in order to achieve desired outcomes in this paradigm. The outcomes are then combined together, harmonized, evaluated. In this paradigm, data ingestion, processing, analysis, and reporting are more complex and challenging processes. Apache Flink and Apache Spark are the most common frameworks used for this paradigm [1], [4]–[6].

In this study, considering the nature of real-time big data processing question, the fundamental approaches of the big data processing are briefly explained, and real-time big data processing is investigated in a comprehensive manner. In this context, lifecycle, tools and tasks and challenges of real time big data processing are revealed and discussed following sections.

The remainder of this paper is organized as follows. Section 2 describes the key concepts of real-time big data processing. Section 3 illustrates the lifecycle of real-time big data processing. Section 4 reveals the tools and tasks of real-time big data processing. Section 5 discusses a number of open issues and challenges. Finally, Section 6 concludes this paper.

978-1-5386-4184-2/18/$31.00 ©2018 IEEE

## II. Real Time Processing of Big Data Streams

A data stream is a continuous, real-time, and unbounded series of data items. The process of streaming divides non-stop flowing input data into distinct units for advanced processing. Stream processing is a low-latency processing approach and analyzing of streaming data. Stream processing is about real-time processing of nonstop streams of data in a workflow. Real-time processing, stream processing, and streaming processing are often used interchangeably.

Real-time processing requires the continual and sequential transactions for limitless streams of input data. In real-time processing, it is essential that the big data streams be processed with very short latency, measured in seconds or milliseconds. So as to accomplish very short latency, big data streams are processed as small sets of input data stored in memory. In other words, the real-time processing can also be stated as a sequence of repeated operations, in which the data streams transferred to the memory instead of the disks are processed as small sets [1], [4]–[6].

Real time processing is crucial in order to continue the high-level functionality of nonstop operated or automated systems having intensive data streams. Numerous platforms, applications, and tools necessitate real-time processing of big data streams having different data structures. Radar systems, smart cities, disaster management systems, internet of things, bank ATMs, and social networks are remarkable examples for the applications of the real-time processing [1], [4]–[6].

## III. Lifecycle of Real-time Big Data Processing

In today's big data platforms, in general sense, big data processing consists of following stages: data acquisition, data storage, data analysis, and data reporting. The organization and application of these phases may vary depending on the background of the data processing model. Big data lifecycle can be applied in various ways for batch processing or real-time processing.

In terms of real-time big data processing, the lifecycle consists of phases with continuity and time-limited. Real time processing focuses the big data streams that are ingested in real-time and processed with minimal latency. Since data stream is continuous in this paradigm, the phase of data storage is usually implemented at the end of the lifecycle. Contrary to the common big data lifecycle, the real-time processing lifecycle for big data streams has five conceptual phases, including real-time data ingestion, data storage, stream processing, analytical data store, and analysis and reporting. The lifecycle is given in Fig. 1 with related tools and tasks.

- *Data ingestion:* This phase is the process in which big data is ingested from heterogeneous data sources. In data ingestion process, lots of real-time processing paradigms use a message ingestion store so as to act as a buffer for messages. In contrast batch processing systems, real-time ingestion can available a stream of data. The data stream is started and continued systematically [5], [7].
- *Data storage:* This phase covers the operations for storage of real-time data streams having different data structures. In the majority of real-time big data applications, the data storage operations are also performed as the analytical data storage at the end of the stream processing [5], [7].
- *Stream processing:* After the data ingestion, real-time big data streams are processed and structured for real-time analytics and decision-making. In this phase, various frameworks and paradigms are used according to the nature of the real-time applications [5], [7].
- *Analytical data store:* In a big data lifecycle, analytical data store is often needed in order to store and serve processed data in a structured format. This process makes it also possible to query the data using analysis tools [5], [7].
- *Analysis and reporting:* In this last phase of the big data lifecycle, it is aimed to provide implications and insights for effective decision-making through analysis and reporting [5], [7].

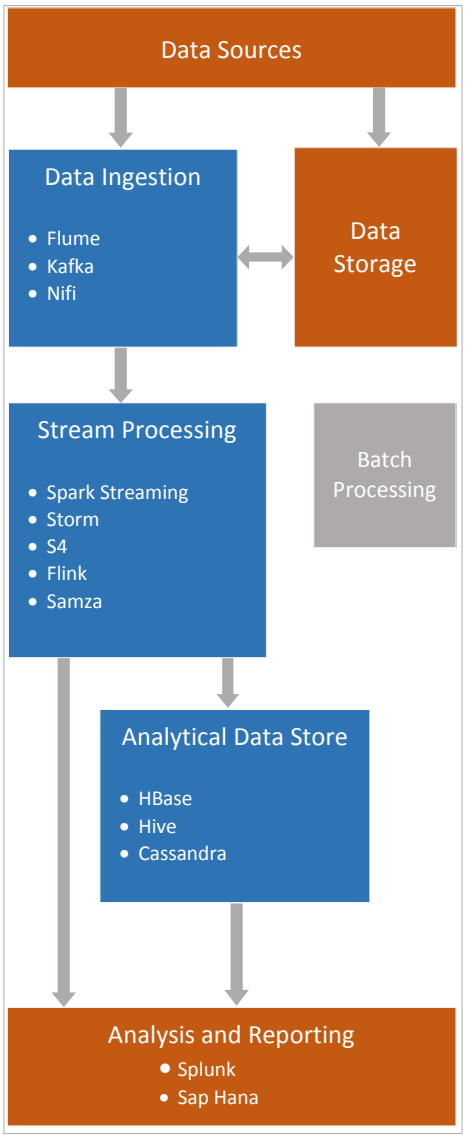


Fig. 1. Lifecycle of real-time big data processing

In the subsequent section, the real-time processing tools and tasks are associated with the big data lifecycle and are then clustered as the sequential phases of the lifecycle.

## IV. Tools and Tasks for Real-time Big Data Processing

This section analyzes the frameworks (platforms, tools, and technologies) are used in the real□time processing of big data streams. Taking into account the processes in the lifecycle, the most common 11 frameworks are discussed and their roles and tasks in the lifecycle are identified. And finally, these tools and tasks are compared, and given in Table 1.

### A. Data Ingestion

- *Flume:* Apache Flume is a reliable, distributed, and obtainable data ingestion system for gathering, combining and transferring large volumes of data streams such as events, log data (etc...) from many distinct data sources to a centralized data store. It has a simple and modular architecture that provides the online analytic application for data streams [5].
- *Kafka:* Apache Kafka is a distributed streaming framework having three main capabilities: First, publish, distribute, and subscribe the streams of records like a message queue or real-time messaging organization. Second, store data streams with fault-tolerant is a robust approach. Third, process streams of events and logs. Kafka delivers reliable and low latency responses to support real-time applications and to connect streaming data systems. Kafka can be used for real-time event processing and integration of modules of large-scale software systems. Compared to Flume, Kafka provides better scalability and message consistency [1], [5].
- *Nifi:* Apache Nifi (short for Niagara Files) is a real-time integrated data streams and simple event processing platform that automates the flow of data streams between disparate data sources and software systems. Unlike Flume and Kafka, Apache Nifi can handle messages with arbitrary sizes. It uses a web-based user-interface having drag-and-drop module and delivers a real-time control that makes it easy to manage the flow of data streams between data sources and the systems [1], [5], [8].

### B. Stream Processing

- *Storm:* Apache Storm is an advanced big data computation framework that allows the real-time processing and distributed computation of data streams at a high speed measured in milliseconds. It is designed to process huge volume of data in a fault-tolerant and horizontal scalable methodology. Storm can be easily integrated and implemented with any programming language. Due to its flexible features, Storm continues to be a leading paradigm in real-time processing of big data streams [1], [5].
- *Spark streaming:* It is a Spark technology for real-time processing, which is named Spark streaming, similar to Spark for batch processing. Spark streaming makes it easy to build scalable streaming applications of live data streams with fault-tolerant. It delivers the real-time processing similar to a sequence of very short batch tasks. In Spark streaming paradigm, data streams are received as live input and separated the data into small batches. After that, the batches are processed by the Spark engine in order to generate the final stream of data outcomes in batches [5].
- *S4:* Apache S4 is a distributed computing and real-time processing platform for unbounded streams of data. S4 provides a general-purpose, distributed, partially fault-tolerant, scalable, extensible, pluggable, and real-time platform in which programmers can easily develop real-time applications for processing continuous data streams. It is initially developed by Yahoo in 2010. The fundamental platform of S4 is developed using Java. S4 has continued an Apache Incubator project since 2011 [1].
- *Flink:* Apache Flink is an open-source streaming data processing tool developed for high-performing, always-available, and distributed implementations. Apache Flink can also handle batch tasks as well as process data streams. It employs batches to be simple data streams with predetermined restrictions, and so, batch processing is employed as a sub-unit of stream processing. The core infrastructure of Apache Flink is based on a distributed dataflow engine developed in Java and Scala. It runs arbitrary streaming data programs in a pipelined, distributed, and data-parallel way. The pipelined runtime system of Flink allows the execution of bulk/batch and stream processing applications [9].
- *Samza:* Apache Samza is an open-source near real-time, asynchronous computational framework for event / stream processing developed in Java and Scala. Samza is a stream processing platform that is strongly based on the Apache Kafka messaging system. It is organized exactly to take advantage of Kafka's unique architecture and guarantees. Samza also employs Apache Hadoop Yarn to make available fault tolerance, and Kafka for messaging, buffering, and state storage [10].

### C. Data Storage / Analytical Data Store

- *Hbase:* Hbase is a distributed and column-oriented database system constructed on top of the Hadoop distributed file system (HDFS). It is an important part of the Hadoop ecosystem that delivers arbitrary real-time read/write access to data in the HDFS. It also makes available a fault-tolerant way of storage large amounts of sparse data. Hbase includes in-memory, operation compression, and bloom filters on a per-column. HBase is now executing numerous data-oriented websites, containing Facebook's messaging platform. Contrary to traditional and relational databases, Hbase does not support SQL scripting; as an alternative the equivalent is written in Java, using similar to a Mapreduce application [1], [5], [11].
- *Hive:* Hive is a data warehouse infrastructure and arrangement system in order to process and analyze structured data in Hadoop. It is commonly used to summarize big data, and make querying and investigating easily. Hive is designed for OLAP (Online analytical processing). It is also a language for real-time queries and row-level updates. It

provides SQL-based language for querying known as HiveQL or HQL [1], [5], [11].

- *Cassandra:* Apache Cassandra is a NoSQL database that provides an open source, distributed and decentralized/distributed storage system for managing large volumes of structured data spread out across the world. It is a column-oriented database that features scalable, fault-tolerant, and consistent. Cassandra delivers robust support for data clusters spanning multiple datacenters with asynchronous masterless replication tolerating low-latency processes for all clients. Especially in recent times, Cassandra is being effectively used by some big corporations such as Facebook, Twitter, Cisco, Ebay, in real-time big data streaming applications. [1], [5], [11].

### D. Analysis and Reporting

- *Splunk:* Splunk is an intelligent and real-time big data platform that captures, indexes, and correlates real-time data in a searchable warehouse from which it can generate dashboards, visualizations, graphs, reports, and alerts. Splunk also provides a real-time data analytics framework used for application monitoring and management, security and compliance, as well as business intelligence and web analytics. It offers a fully-documented and supported REST API methodology. Application developers can use index, search, and visualize data in [1].
- *Sap Hana:* It is a column-oriented and in-memory relational database management system and database server so as to store and retrieve data requested by the real-time applications. Besides, Sap Hana achieves advanced real-time analytics of big data streams (prescriptive and predictive analytics, sentiment analysis, spatial data processing, streaming analytics, text analytics, social network analysis, text search, graph data processing) and contains ETL capabilities as well as a real-time application server [1].

TABLE I. COMPARISON OF TOOLS USED FOR REAL-TIME BIG DATA PROCESSING

| *Name* | *Specified use* | *Supported languages* | *Primarily written in* |
|---|---|---|---|
| Flume | Data ingestion, Single-event processing | Java, Scala, Python, R | Java |
| Kafka | Data ingestion, Event stream processing | Java, Scala, Python, Ruby, R, Clojure | Java, Scala |
| Nifi | Data ingestion , Single-event processing | Java, Python, Clojure, JavaScript, Groovy | Java |
| Storm | Event stream processing, Complex event processing | Java, Scala, Clojure, Python, Ruby, C# | Clojure |
| Spark Streaming | Event stream processing, | Java, Scala, Python, R | Scala |
| S4 | Event stream processing, Complex event processing | Java, C++, Python, Clojure | Java |
| Flink | Event stream processing, Complex event processing | Java, Scala, Python, R | Java |
| Samza | Event stream processing | Java, Scala, Python, Ruby | Java, Scala |
| HBase | Wide-column store based on BigTable | Groovy, Java, PHP, Python, Scala, C++ | Java |
| Hive | Data warehouse for large-distributed datasets | Java, PHP, Scala, C++ | Java |
| Cassandra | Wide-column store based on BigTable and DynamoDB | C#, Clojure, Python, Go, Java, Perl, PHP | Java |
| Splunk | Analytics platform | C#, Java, JavaScript, PHP, Python, Ruby | Python, C++ |
| Sap Hana | Column based store, Analytics and application platform | SQLScript, R | C++ |

## V. CHALLENGES

In modern day environments, big data is mostly described by its main characteristics such as large volume, variety, and heterogeneity of data in which all processes require new technologies and methodologies in order to extract valuable information from it. In this phase, the essential challenges discussed in this study were grouped under three main categories: characteristic challenges, processing challenges, and management challenges, as shown in Fig. 2. In the rest of the article, the challenges are discussed in more detail and specified under eight distinct headings.

### A. Volume, Variety and Heterogeneity

Unstructured data can contain all types of data in high volume and different formats. These data may include different information sharing platforms such as social media, forum, e-mail, chat, online communities, online shopping, and so on. Moreover, the data may include a combination of different file formats such as picture, video, audio, and text. The analytical processes on the data are more difficult and costly due to this complex structure of the data. This heterogeneous and complex structure of big data is a serious challenge that must be overcome in big data analytics process.

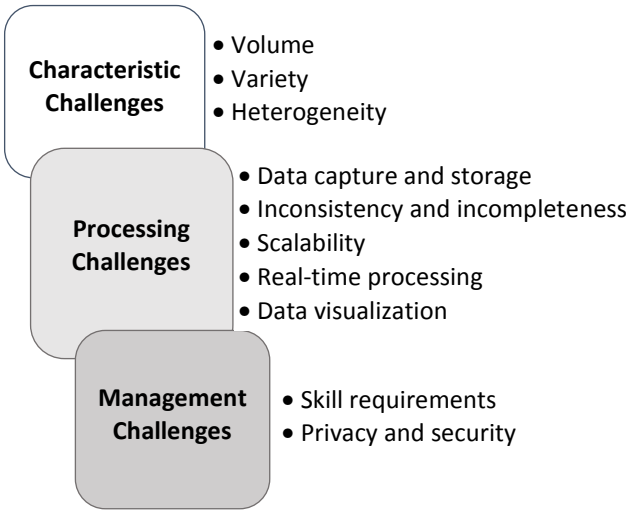


Fig. 2. An overview of the big data challenges

The available traditional data processing methods are only implemented on homogeneous and structured data. Therefore, considering the complex structure of big data, it requires the development of innovative data processing methods. In addition, the heterogeneous nature of big data

should be carefully structured and converted to homogeneous data for processing by traditional analysis methods. The data processing systems and big data sources can be combined on a common platform and thus the data can be provided to be in a more appropriate structure for these analysis processes [1], [3], [12], [13].

### B. Data Capture and Storage

In today's digital environments, it is generated daily 2.5 quintillion bytes of data, and this amount is ever-increasing day by day. The collection and storage of the data in this very large volumes can be performed by various technical processes that require large costs. Several log records are periodically deleted in different data sources, because of the cost of required storage systems. The advent of big data models changes the traditional data capture and storage methodologies, data access systems, and data-driven implementations. Because existing storage technologies cannot provide to necessary performance in terms of real-time data collection, processing, and storage. Big data processing and analytics requires high-speed data input/output (I/O) access patterns, because these processes cannot be achieved with current hard disk drives (HDDs) based on random I/O access. In this regard, solid-state drive (SSD) and phase-change memory (PCM) technologies using sequential I/O patterns are mostly preferred for big data storage instead of traditional HDDs. In order to overcome this challenge, innovative storage technologies such as direct-attached storage (DAS), network-attached storage (NAS), and storage area network (SAN) were began to be used in order to achieve advanced storage processes [1], [3], [12], [13]. Furthermore, distributed, parallel, and cloud-based data centers provide a new solution approach to data capture and storage in approved manner.

### C. Inconsistency and Incompleteness

Big data is a combination of data obtained from different data sources in different formats. In this case, data inconsistency and incompleteness may occur in the processing of data obtained from different source, and so data processing may not be achieved in desired level. Considering the heterogeneous nature of big data, at the start, the reliability of the data collected from different sources should be increased by verifying as temporal and spatial. After that, a common homogeneous data structure should be determined and the data obtained from different sources should be combined on monotype basis. Also, data transfer process must be checked regularly to ensure whether the process is complete or not. Thus, the inconsistency and incompleteness errors may be controlled during big data processing [1], [3], [12], [13].

### D. Scalability

Recently, the optimization of rapidly growing data emerges a challenging process for providing scalability, a remarkable challenge should be solved in up-to-date database systems. It is thought to be a scalable system, whether the system's performance and capacity relatively growths after installing a hardware or application. Big data scalability is based on the essentials of scaling databases from a single node to large clusters [13]. One process employed by most of the major database management system (DBMS) applications is the partitioning of large tables, based on arrays of values in a key field. In this way, the databases can be scaled out across a cluster of distinct big data servers. Furthermore, with the advent of 64-bit multi-core CPUs, DBMS providers have focused on multi-threaded implementations that essentially scale up transaction processing capacity. Storage area networks (SANs), network-attached storage (NAS) and direct-attached storage (DAS) together with fast local area networks empower the structures of big databases and distributed computing processes [1], [3], [12], [13]. In the scaling of data, the usage of cloud computing processes is ever-increasing each passing day. As a result, providing of synchronization between cloud computing processes is one of the main tasks in big data processing.

### E. Real-Time Processing

As volume and variety of data increase, timeliness on big data processing becomes a more complicated challenge. The timeliness can be defined as the time delay between generating and ingestion of data. Data should be obtainable in this period in order to achieve for an effective analytics. For example, in a traffic monitoring system watching millions of vehicles instantly, finding the alternative ways and calculation of the arrival times require the real time data processing. In this respect, any delay or miscalculation may misdirection an ambulance, in this aspect, timeliness has a vital importance in terms of human life. Thus, timeliness is one of the most essential measurements of data quality because of the increasing demand for real-time decision support systems. In real-time processing, it is very important that providing the necessary time to perform the needed operations on large-volume data streams. Moreover, the timeliness of big data may facilitate the processing of event streams in order to provide real-time decision making. The heterogeneous data sets generated from various data sources should be combined into an analytical platform in order to minimize the potential time lags for real-time processing [1], [3], [12], [13].

### F. Data Visualization

Data Visualization is used to report information clearly and effectively to users by employing different presentation techniques such as graphs, tables, charts, and animations. It can be helpful for users to understand and interpret the great amounts of processed data in a simpler way. Big Data visualization is not as easy as traditional small data sets. In complex and large-scale data sets, the visualization process requires many innovative approaches and components. Determination of the most suitable data presentation model is crucial in order to achieve an understandable visualization. The progression of traditional visualization approaches is relatively remarkable, but not at the desired level. In large-scale data visualization, many generative techniques such as feature extraction and geometric modeling are suggested for data visualization and presentation. Thus, data visualization approaches should be redesigned in light of big data trends and demands. To begin with, big data visualization should provide an effective overview, afterward a flexible scaling and filtering. The visualization process should be carried out to make the large-scale data more manageable, understandable, and functional [1], [3], [12], [13].

### G. Skill Requirements

The big data industry is an emerging and innovative job market in which labor resources are employed effectively. In

this competitive environment, in-demand skills, competencies, and requirements are ever-changing and progressing. Big data specialists are expected to have a wide spectrum of knowledge and skills together big data labor market. For this reason, big data specialists should always progress their knowledge and skills up to date. As big data industry advances and business demands increase, new career opportunities are opening up every day for big data specialists, and therefore skilled labor shortage is ever-increasing in these days. The scarcity of big data talent will be a significant challenge in order to meet emerging market demands in this field. According to the latest statistics, by 2018, the United States alone could face a scarcity of 140,000 to 190,000 employees with deep analytical knowledge and skills as well as 1.5 million managers, analysts, and specialists used the big data analytics results to make effective decisions [14]. In this developing field, the training of big data specialists with a strong technical background is a remarkable challenge should be achieved [1], [3], [12], [13].

### *H. Privacy and Security*

Some specific information such as individual health information, online shopping memberships, and bank account numbers, and social media profiles require a high level of protection in terms of privacy and security. In this regard, the most important threat for the security of personal information is that the personal data is irregularly accumulated by the numerous social media platforms. Undoubtedly, specific big data resources must be safeguarded by laws and regulations in terms of privacy and security. International Data Corporation (IDC) recommended five levels of increasing security: privacy, compliance-driven, custodial, confidential, and lockdown. The private personal information of a person when combined with that person's shares provide the significant facts and details about the person. The discovery of the specific personal information is particularly supportive for criminal investigations. On the other hand, people may not want to be shared or used their personal information by the others [1], [3], [12], [13]. For this reason, in the subject of privacy and security of the data used in big data analytics, there are needed additional studies and regulations that reveal the required security levels.

## VI. Conclusion

This paper provides a brief overview on real-time processing of big data streams, with its lifecycle, tools and tasks and challenges. This paper initially revealed the lifecycle of real-time big data processing, consisting of four phases, that are data ingestion, data processing, analytical data store, and analysis and reporting. Secondly, it described tools and tasks of real-time big data processing. These tools are: Flume, Kafka, Nifi, Storm, Spark Streaming, S4, Flink, Samza, Hbase, Hive, Cassandra, Splunk, and Sap Hana. And finally, challenges of real-time big data processing were identified and categorized. The challenges are: “volume, variety and heterogeneity”, “data capture and storage”, “inconsistency and incompleteness”, “scalability”, “real-time processing”, “data visualization”, “skill requirements”, and “privacy and security”.

This paper may provide valuable insights into: 1) companies, in employing a qualified big data workforce and in integrating new big data paradigm into evolving business strategies; 2) big data professionals, in assessing and improving their own qualifications; 3) academic communities, in designing big data programs and curricula in line with emerging trends and technologies.

Entering the era of big data, in order to achieve more effective processing and analytics of big data streams and to take advantage of the opportunities of big data realm, the current approach and paradigms of big data processing should be analyzed in a comprehensive manner. To achieve this, it seems that more research and applications are needed about processing of big data streams effectively. The researches and applications can provide valuable contributions to the understanding of the main characteristics and tendencies about the big data processing.

## References


[1] C. L. Philip Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on Big Data,” Inf. Sci. (Ny)., vol. 275, pp. 314–347, 2014.

[2] D. Laney, “3D data management: Controlling data volume, velocity and variety.,” META Gr. Res. Note, vol. 6, no. February 2001, p. 70, 2001.

[3] Z. Zheng, P. Wang, J. Liu, and S. Sun, “Real-Time Big Data Processing Framework: Challenges and Solutions,” Appl. Math. Inf. Sci, vol. 9, no. 6, pp. 3169–3190, 2015.

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

[5] R. Casado and M. Younas, “Emerging trends and technologies in big data processing,” Concurr. Comput. Pract. Exp., vol. 27, no. 8, pp. 2078–2091, 2015.

[6] A. A. Safaei, “Real-time processing of streaming big data,” Real-Time Syst., vol. 53, no. 1, 2017.

[7] “Real time processing|Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/real-time-processing#technology-choices. [Accessed: 08-Jun-2018].

[8] B. Samal and M. Panda, “Real Time Product Feedback Review and Analysis Using Apache Technologies and NOSQL Database,” Int. J. Eng. Comput. Sci., vol. 6, no. 10, pp. 22551–22558, 2017.

[9] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, “Apache flink: Stream and batch processing in a single engine,” Bull. IEEE Comput. Soc. Tech. Comm. Data Eng., vol. 36, no. 4, 2015.

[10] G. Hesse and M. Lorenz, “Conceptual survey on data stream processing systems,” in Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on, 2015, pp. 797–802.

[11] Y. Shi, X. Meng, J. Zhao, X. Hu, B. Liu, and H. Wang, “Benchmarking cloud-based data management systems,” in Proceedings of the second international workshop on Cloud data management, 2010, pp. 47–54.

[12] A. Katal, M. Wazid, and R. H. Goudar, “Big data: Issues, challenges, tools and Good practices,” in 2013 6th International Conference on Contemporary Computing, IC3 2013, 2013, pp. 404–409.

[13] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. Mahmoud Ali, M. Alam, M. Shiraz, and A. Gani, “Big Data: Survey, Technologies, Opportunities, and Challenges,” Sci. World J., vol. 2014, pp. 1–18, 2014.

[14] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, “Big data: The next frontier for innovation, competition, and productivity.” May-2011.