IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 September 2023, accepted 6 October 2023, date of publication 11 October 2023, date of current version 17 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323160

RESEARCH ARTICLE

A Comparative Perspective on Technologies of Big Data Value Chain

AHMET ARIF AYDIN
Computer Engineering Department, Inonu University, 44280 Malatya, Turkey
e-mail: arif.aydin@inonu.edu.tr

ABSTRACT Data is one of the most valuable assets in the digital era because it may conceal hidden valuable insights. Diverse organizations in diverse domains overcome the challenges of the big data value chain by employing a wide range of technologies to meet their needs and achieve a variety of goals to support their decision-making. Due to the significance of data-oriented technologies, this paper presents a model of the big data value chain based on technologies used in the acquisition, storage, and analysis of data. The following are the paper's contributions: First, a model of the big data value chain is developed to illustrate a comprehensive representation of the big data value chain that depicts the relationships between the characteristics of big data and the technologies associated with each category. Second, in contrast to previous research, this paper presents an overview of technologies for each category of the big data value chain. The third contribution of this paper is to assist researchers and developers of data-intensive systems in selecting the appropriate technology for their specific application development use cases by providing examples of applications and use cases from prominent papers in a variety of fields and by describing the capabilities and stages of the technologies being presented so that the right technology is used at the right time in the big data collection, processing, storage, and analytics tasks.

INDEX TERMS Analytics, acquisition, big data technologies, information systems, storage.

I. INTRODUCTION

In today's digital world, an exponential growth of data has been triggered by the production of well-equipped portable devices and availability of affordable smart mobile devices, ubiquitous internet access provided by GSM companies' cellular networks, development and numerous usages of web services in almost all sectors, accessibility of these services through internet and dramatic changes in billion people's habits and daily activities. According to an infographic provided by DOMO Company [1], every minute of a single day, enormous amounts of data have been generated by a variety of sources [2], [3], [4]. For example, in every minute of a single day, Twitter users send 575K tweets, Google conducts 5.7 million searches, ZOOM hosts 856 minutes of webinars, Netflix customers stream 452K hours of videos, and YouTube users' stream 694K hours of videos. These examples represent just a small portion of data generated and managed by well-known companies worldwide [5], [6]. Moreover, the coronavirus pandemic has also forced people to generate large amounts of data with performing activities online, such as working from home, distance education, shopping, socializing through web platforms, playing games, video conference meetings, academic conferences, or performing medical operations. These examples indicate that almost everyone around the world has been involved in the generation of data.

Data is called the new oil [7] because it may contain hidden valuable insights [8]. Thus, organizations of diverse domains have been involved in collecting data from numerous sources, storing it by utilizing more than four hundreds of data storage technologies [9] that are relational database management systems (RDBMS), NoSQL, NewSQL, and other storage technologies, and analyzing data in motion with streaming processing technologies or utilizing batch style processing tools to analyze data at rest to accomplish a variety of purposes in diverse domains. For example, in the business world, companies collect their customer's purchasing data,

The associate editor coordinating the review of this manuscript and approving it for publication was Abdel-Hamid Soliman.

feedbacks, and comments on products to recommend a viable product, predict future trends, and provide better services for them to stay in the job market and compete with their rivals. On the other hand, gleaning beneficial information and insights out of the new oil does not come for free; therefore, organizations must deal with numerous challenges associated with collecting, processing, modeling, storing, managing, and analyzing stages of big data to accomplish their goals.

In this paper, due to the importance of big data value chain and technologies, a big data value chain-oriented perspective is presented on big data technologies utilized in acquiring, storing, and analyzing data. This paper first provides a background on presenting the concept of big data is by providing definitions, characteristics, big data processing paradigms, and data analytics types. And then, a comparative perspective on selected technologies is presented regarding the big data value chain, usage purposes, and capabilities. Last, use cases of big data applications in diverse domains are presented to exemplify usage of big data technologies in various domains. This research aims to support researchers and developers of data-intensive systems in their development efforts to use the right technology at the right stage in the big data value chain based on published research endeavors from academia and industry.

The organization of the paper is as follows. In Section II, definitions, characteristics, data processing paradigms, and data analytics types of big data are presented. In Section III, the methodology of this paper is presented. Section IV provides a model of big data value chain. Section V provides comparative perspective on big data technologies regarding the big data value chain. Section VI presents related works. Section VII presents selected use cases of big data application examples and usage purposes of various domains. Section VIII provides a discussion and limitations. In Section IX, a conclusion is provided by summarizing the contributions of this paper.

II. BIG DATA CONCEPTS AND TERMINOLOGY

This section provides a background on definitions and characteristics of big data, data processing paradigms, and data analytics types.

A. DEFINITIONS OF BIG DATA

Define abbreviations The big data term was initially introduced in [10], to explain visualization challenges. According to the authors, ''the visualization provides an interesting challenge for computer systems as the - data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk.'' This definition provided insight into computer systems at that time but was insufficient to handle big data challenges. Moreover, big data is considered ''massive data,'' ''heterogeneous,'' ''unstructured,'' and ''very large data'' [7] that cannot be handled by using traditional technologies [11]. Following big data definitions are presented from prominent publications:

- In McKinsey's report: ''Big data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze'' [12].
- In IDC's report: ''big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis'' [13].
- ''Too big, too fast, or too hard for existing tools to process'' [14].
- ''Big data is where the data volume, acquisition velocity, or data representation limits the ability to perform effective analysis using traditional relational approaches or requires the use of significant horizontal scaling for efficient processing'' [15].
- ''Big data is a term for massive data sets having a large, more varied and complex structure with the difficulties of storing, analyzing and visualizing for further processes or results'' [16].
- ''Big data consists of extensive datasets - primarily in the characteristics of volume, variety, velocity, and/or variability- that require a scalable architecture for efficient storage, manipulation, and analysis'' [17].
- Gartner's glossary: ''Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation'' [18].

Furthermore, a comprehensive list of big data definitions is provided in [17], and each explanation is focused on different aspects that are volume, velocity, variety, value, new data types, big data engineering, analytics, bigger data, sampling, data science, and cultural change.

B. CHARACTERISTICS OF BIG DATA

Regarding the preceding definitions, widely accepted characteristics of big data are volume, velocity, variety, and value [19], [20], [21]. Volume is about the size of large amounts of data in Gigabytes, Terabytes, Petabytes, Exabytes, Zettabytes, or Yottabytes [7]. Due to large volumes, data cannot be stored, managed, and analyzed with traditional computing and storage systems. Instead, contemporary technologies are utilized to collect, store, manage and analyze big data. Velocity is the speed of incoming data from a source that needs to be captured, processed, and transferred quickly [14]. Variety is about various types of data in different formats, such as structured, semi-structured, and unstructured [3], [6], [22]. Value is a vital characteristic of big data that is about deriving beneficial information and hidden insight out of a large amount of data using various data mining and machine learning techniques and algorithms [23], [24].

Beyond the mentioned 4V's of big data, veracity, variability, and valance characteristics are also mentioned in big data studies. Veracity is considered the messiness or trustworthiness of data [6], [25]. Veracity impacts the result of various analyses because of the quality, objectivity, credibility, and accuracy of data [23], [22], [26], [27]. Variability is about dynamically and rapidly changing data flow at unpredictable rates [3], [23]. Valance is about the connectivity of data [24].

C. DATA PROCESSING PARADIGMS

In the big data processing context, data processing paradigms are classified under two categories that are batch and streaming processing [28], [29], [30].

The batch processing paradigm aims to process stored data (data in rest) to perform exploratory and detailed analysis on complete datasets by processing every item in a dataset to calculate answers for the analyst's predetermined set of questions via making use of statistical analysis, data mining, and machine learning. Popular batch processing tools are presented in Section V. Moreover, processing big data in reasonable amounts of time with traditional ways is almost impossible. To address this problem and decrease the time amounts of data processing, the MapReduce programming model was developed by Google to utilize parallelism in multi-core and many-core clusters [31]. MapReduce programming model enables programmers to create a map and reduce functions. Multiple map functions are launched to process large amounts of data in parallel. Each map function receives a small portion of data in key-value pairs, performs user-written operations, and produces partial (intermediate) results. Reduce function is also user-defined and receives intermediate results from the map function. Also, multiple reduced functions can be utilized to consolidate intermediate results. Due to its efficiency and ease of use, the MapReduce programming model has created a basis for data processing technologies [32] presented in Section V.

Stream processing aims to capture, manage, and process data in motion (fast data) from data sources before permanently storing it [7]. This paradigm requires capturing, filtering, and processing data in short amounts of time (seconds, milliseconds) to take advantage of the freshness of data that contains a potential value. Real-time or near real-time analytics is crucial in today's big data world to provide fast answers for various domains [33]. Fig. 1 represents where stream processing can be performed in big data processing stages and which tools can be used to perform streaming analytics that are explained in Section IV.

D. DATA ANALYTICS TYPES

In big data analytics, data analytics can be classified under the following categories: descriptive, diagnostic, predictive, and prescriptive [4], [34], [35], [36], [37] that, are related to value characteristics of big data and expected value increases moving from descriptive toward prescriptive analytics. In these types of analytics, batch-oriented detailed exploratory analytics are performed on a large amount of previously collected data to glean beneficial information.

Descriptive analytics focuses on the "What happened?" question to glean information from previous activities and events to figure out successes and failures and learn from an organization's history [34], making use of statistical techniques, dashboards, BI tools, and visual representations. Moreover, descriptive analytics can be considered "hindsight" since it enables one to gain experience by learning from past activities. For example, a company would like to know the number of users who unsubscribed to their email during the last five months.

Diagnostics analytics aims to answer the "Why did an event happen?" question via detailed investigation and interactive data visualization tools to understand which factors caused or triggered a particular event [35]. Diagnostic analytics can also be considered as "hindsight" since it aims to figure out hidden insights from large amounts of data. For example, a company would like to determine information about its workers' significant performance loss during summertime.

Predictive analytics strive to answer, "What will happen in the near future?" with the goal of predicting prospective results of an action in advance. Predictive analytics provides "insights" before an event occurs, such as customer behavior prediction, energy consumption, or amount of income or loss. Moreover, predictive analytics utilizes machine learning, statistical methods, and data mining algorithms to forecast the implications of a future event [4].

Prescriptive analytics tries to answer the "How can we make it happen?" question using statistical optimization techniques, artificial neural networks, simulations, expert systems, and game theory [34]. Prescriptive analytics can be considered "foresight" since it recommends actions to accomplish beneficial future results. For example, a company would like to forecast about impact of their new product on the sale market.

III. METHODOLOGY

Big data processing, storage, and analytics technologies are widely used in almost all domains. Due to various demands and requests from diverse domains, a large number of technologies have been developed. Therefore, in the methodology of this paper, at most five big data technologies are chosen and presented in each stage of the big data value chain because this paper aims to present technologies for each category of the big data value chain, and some categories, such as storage technologies, include more than four hundred technologies. Thus, to accomplish the goal of presenting technologies from each category, at most five carefully chosen technologies are presented in each category. The selection criteria for chosen technologies are becoming open source, freely available, and utilized in a data-intensive system that appears in published research works that are indexed in the Web of Science.

Moreover, demanded features from technologies of acquisition, temporary storage, permeant storage, streaming analytics, and batch analytics are different from each other since each stage has its own focus, requirements, and goals. Therefore, features of big data technologies for each stage of the comparison of the big data value chain are carefully selected based on the requirements and goals of each category. And then, a fair comparison is objectively provided on the

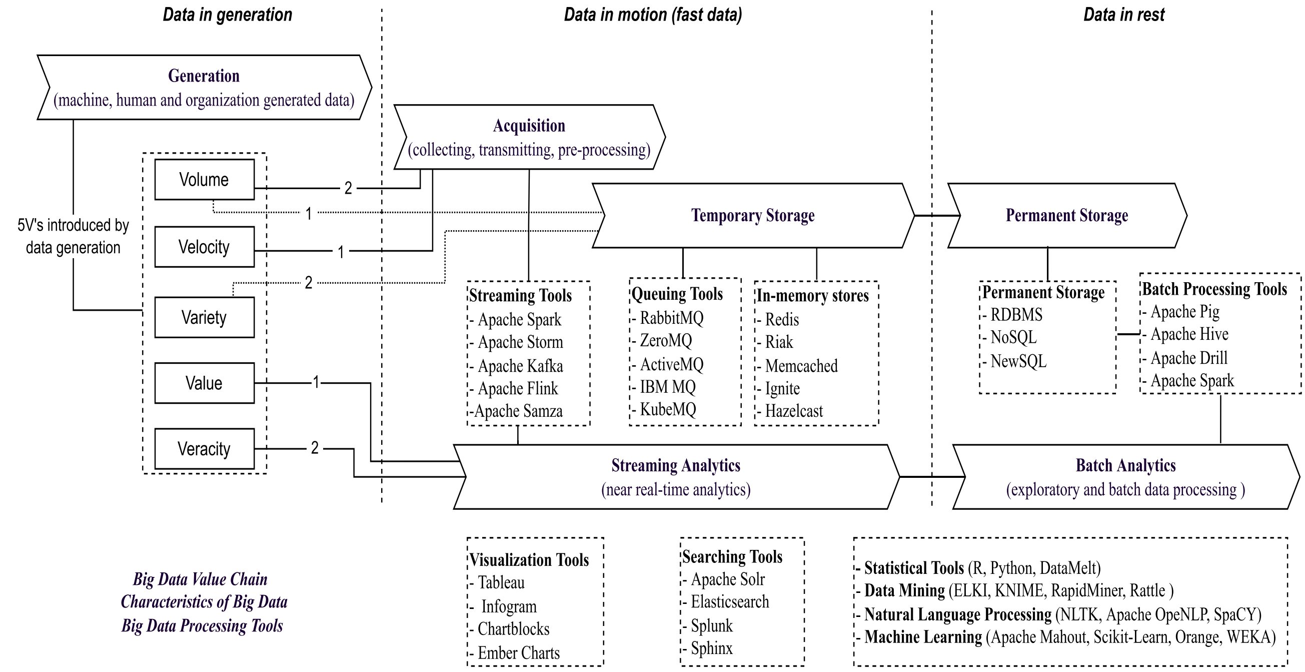

FIGURE 1. Stages of big data value chain model, 5V's of big data and utilized technologies.

capabilities of each technology, with the purpose of giving insight and a feature overview of the presented technologies for researchers and developers in data-intensive systems and big data analytics.

In addition, in the context of big data and analytics, hundreds of papers have been published. However, the papers cited in this work and included in the related works section were chosen based on the following criteria: including "big data value chain," "big data technologies," 'acquisition technologies," "storage technologies," and "analytics technologies" terms on a Web of Science search, and focusing on big data technologies that are utilized stages of the big data value chain.

Finally, this paper aims to present a comparative study of frequently utilized big data technologies in the generation, acquisition, storage, and analytics stages of the big data value chain. This paper is written to answer the following research questions, each of which is addressed in the upcoming sections:

- RQ1: How can a big data value chain model be created with the purpose of presenting characteristics of big data, usage of big data processing tools, and relations between generation, acquisition, storage, and analytics stages of the big data value chain?
- RQ2: What are the significantly important features to compare various technologies used in stages of the big data value chain?
- *RQ3:* How can we provide a comparative perspective on big data technologies used in stages of the big data value chain to support researchers in big data and analytics?

IV. BIG DATA VALUE CHAIN MODEL

In this section RQ1 is addressed. Regarding notable big data research work [7], [19], [38], [39], big data value chain is typically represented under *generation*, *acquisition*, *storage*, and *analytics* stages. These stages are generally represented separately but these stages of the big data value chain should not be strictly separated from each other. Thus, Fig. 1 is depicted to represent the relations among stages of the big data value chain, characteristics of big data, and stage-related popular technologies and tools. Moreover, to the best of my knowledge, characteristics of big data are generally represented standalone in big data research [34], [40], [41]. However, in Fig. 1, the big data value chain and 5V's of big data are represented all in one place to indicate the relationship among each other.

Generation is about creating data in different types, diverse formats, and speeds from various sources. Data generation is supported by machines, humans, and organizations [24]. First, machines generate data by physical devices and linked software systems such as sensors, video and image capturing cameras, audio recording devices, satellites, smart portable devices (phones, watches, and wearable technologies), computers, cloud systems, transportation systems (airlines, trains, and bus), GPS, IoT devices, and servers. Second, humans around the globe trigger generation of data through various activities with the aid of sources such as social media microblogging services (Twitter, Facebook, LinkedIn, YouTube), news sites, blogs, Internet forums, scientific experiments, education, telecommunication services, customer relationship management (CRM) systems, e-commerce, user feedback, and review pages. Third, organizations generate data through various services such as governmental institutions, healthcare organizations, business services, industrial tasks, manufacturing and retail sector, banking system, and finance. Also, the generation stage introduces 5V's of big data, and the following stages inherit characteristics of big data, but the priority of these characteristics can be different for each stage. For example, velocity challenge is directly connected and related to the acquisition stage which is shown with a flat line with and given number is 1 in Fig. 1.

*Acquisition*involves collecting, transmitting, and pre-processing data created in the generation stage. Thus, this stage deals with data in motion (fast data). The main concerns are acquiring data generated by various sources without losing, pre-processing acquired data to eliminate redundancy and noise to prepare data for storage, and making data available for future analytics. Regarding 5V's of big data, the most important two characteristics in the acquisition stage, as shown in Fig. 1, are *velocity* (speed of incoming data) and *volume* (size of incoming data) since both are very important in capturing and transmitting data. The capabilities of selected acquisition technologies are explained in Section V-B.

Storage focuses on properly storing data in diverse formats received from acquisition tools. In traditional practice, data is permanently stored in RDBMS; however, in the contemporary big data world, in-memory NoSQL data stores and queuing technologies are paramount to store data temporarily before permanently storing data that is a critical role in answering user demands in short times. One of the essential concerns for storage technologies is choosing a proper one or multiple data storage systems out of more than four hundreds RDBMS, NoSQL, and NewSQL technologies [9]. Choosing a proper data storage system for the job is crucial because it is not only about storing data but also easing future demanded analytics on the stored data.

*Analytics*involves getting insights and beneficial information out of data. The primary purpose of collecting tremendous amounts of data is not just increasing volumes in petabytes but also gleaning beneficial information out of data to take meaningful and profitable actions at the right time and in the right place [42]. Thus, in the current view of the big data value chain, streaming and batch data analytics should be considered as represented in Fig. 1. Batch data processing is crucial to perform exploratory data analysis via various statistical, data mining, or machine learning tools. On the other hand, streaming data analytics can be carried out right after acquiring data from various sources to perform real-time analytics by using streaming analytics and temporary storage tools. Fig. 1 provides data processing tools utilized in each stage of the value chain, and these technologies and their features are explained next. Also, Fig. 1 includes examples of visualization, searching, data mining, NLP, ML and statistical tools.

V. BIG DATA VALUE CHAIN AND TECHNOLOGIES

There are many technologies utilized in each stage of the big data value chain, and due to space limitations, only a few selected technologies are presented for each stage. The inclusion of each technology is decided based on their usage in notable published research works that are cite. This section addresses the RQ2 and RQ3.

A. HADOOP ECOSYSTEM

Apache Hadoop is an open-source framework and defacto standard that was developed by Doug Cutting in 2006 to handle the volume challenge of big data via storing data in distributed thousands of nodes using commodity hardware and providing distributed computing and analysis [3], [34].

Apache Hadoop aims to achieve scalability (vertical and horizontal), availability, fault tolerance, and flexibility. Various big data tools have been developed to integrate with the popular Apache Hadoop environment to achieve purposes. For example, managing, monitoring, provisioning, and securing Hadoop Cluster via Ambari, collecting/ingesting data into Apache Hadoop from multiple sources with Flume, Sqoop, and Chukwa, and querying data stored in Hadoop without changing data structures by Hive, Impala, Tajo, and Drill, performing searching demands with Solr, and Lucene, coordinating workflow schedules with Zookeeper, and Oozie, processing data with Pig, Mahout, Spark, Tez, and Flink and storing data in Cassandra, or HBase [2], [43], [44], [45], [46]. Table 1 is formed to provide a comparison among Apache Hadoop versions based on essential features [47]. On the other hand, beyond the open-source versions of Apache Hadoop, commercial Hadoop distributions are also provided by Cloudera, MapR, and Hortonworks.

Due to Apache Hadoop's development purpose, it is not suitable to perform real-time data processing analytics. For example, credit card fraud analytics, network fault prediction, or security threat prediction cannot be detected quickly [48]. Thus, to handle the limitations of Apache Hadoop, various technologies have been developed to provide real-time data analytics and streaming processing that can be integrated with Apache Hadoop environment [2].

B. ACQUISITION TECHNOLOGIES

The main concern in the acquisition stage is collecting, pre-processing, and transmitting data generated from various sources. Thus, acquisition technologies are also considered stream processing tools. This section provides open-source acquisition technologies that are utilized in industry and academic research works.

Acquisition technologies deal with small chunks of continuously incoming data at unknown rates, primarily utilize the underlying machine's limited memory, taking advantage of parallel computations, and performing on the fly operations to provide results of analytics in very short amounts of time in seconds or milliseconds [49]. This time restriction is required in finance, banking, network, and traffic monitoring, fraud detection, or emergency management applications to perform

TABLE 1. Comparison of apache hadoop versions.

Features	**Hadoop 1.x**	**Hadoop 2.x**	**Hadoop 3.x**
Components	HDFS, MapReduce (Batch processing and Resource Management)	HDFS, MapReduce 2 (Batch processing), YARN 1 (Resource Management)	HDFS, MapReduce 2, YARN 2
Port Range	Linux ephemeral port range	Linux ephemeral port range	Moved out ephemeral port range
Data Balancing	HDFS balancer	HDFS balancer	Intra-data node balancer
Fault-tolerance	Replication (3x)	Replication (3x)	Erasure coding
File System Compatibility	Local file system, HFTP, Amazon S3	Azure Storage Blobs, OpenStack Swift, and previous file systems	Aliyun OSS, Azure Data Lake Storage, Tencent COS, and previous file systems
License and Release Year	Apache 2.0, 2011	Apache 2.0, 2012	Apache 2.0,2017
Cluster Size	4,000 nodes	<=10,000 nodes	> 10,000 nodes
Namespace Management	Single Name Node	Single Name Node	Multiple Name Node
OS Support	Linux	Windows, Linux, MacOS	Windows, Linux, MacOS
Storage Overhead	200%	200%	50%

the right actions at the right time. A fair comparison of five well-known acquisition technologies is provided in Table 2.

Apache Spark is one of the most popular open-source data processing engines initially developed by Matei Zaharia in 2009 [50] at UC Berkeley AMPLab and then moved to Apache Software Foundation in 2013 [51]. Apache Spark enables streaming and batch data processing and analytics performance. Resilient Distributed Dataset (RDD) is the fundamental abstraction utilized to manipulate ingested data from various sources and formats [23]. The main reason behind the fast data processing capability of Apache Spark is efficiently using distributed in-memory data structure (RDD) and providing less expensive data shuffles [2]. Thus, the popularity of Apache Spark has dramatically increased since its development, and the demand for using Apache Spark will continue to grow in the future due to its community support, capabilities, and features [52]. Apache Spark also provides libraries for machine learning, fast SQL querying, and data analytics.

Apache Storm is a real-time data processing engine originally developed by Nathan Marz in BackType in 2011 and then open-sourced by Twitter. Apache Storm is a low latency distributed real-time stream processing framework that enables performing analytics before storing the data [23]. Apache storm allows the transformation of one stream to new streams reliably, and it utilizes components *spout* (emits tuple stream) and *bolt* (process tuple and emits a new stream) [53]. A storm cluster can execute one or more worker processes using spouts and bolts. Apache Storm can handle data velocities of tens of thousands of messages every second and is designed to integrate with existing queuing and bandwidth systems. Also, Apache Storm is utilized for real-time analytics, online machine learning, and continuous computation [54].

Apache Kafka is a distributed message system developed at LinkedIn in 2010 to handle streaming activity to process millions of messages per second and then open-sourced by Apache Software Foundation. Apache Kafka provides zero message loss, exactly one processing, and guaranteed ordering; therefore, it is used for mission-critical applications, streaming analytics, and high-performance data pipelines [55]. Apache Kafka's main components are topic (stream of messages), producer (publish messages to a particular topic), brokers (stores published messages), and consumers (gets messages from brokers). Apache Kafka is explicitly distributed, and multiple producers, brokers, and consumers are supported based on usage. All data in Kafka is immediately written to a persistent log on the file system, and each message is addressed by its logical offset in the log without an explicit message ID [23].

Apache Flink is a unified stateful streaming and batch data processing framework developed in 2009 and then incubated in Apache Project in 2014 [53]. The main components of Apache Flink are *stream* and *transformations* [2], [23], [56]. Apache Flink uses a stream-first approach and Kappa architecture (true streams used) to provide automatic partitioning and caching. Moreover, Apache Flink is mainly utilized for event-driven (fraud and anomaly detection, rule-based alerting, business process monitoring), data pipelining (Continuous ETL), and data analytics applications [57].

Apache Samza is a scalable real-time data processing engine that provides streaming and batch data processing. Apache Samza was developed at LinkedIn and then open-sourced by Apache Software Foundation in 2013. Apache Samza manages data streams using *streams and partitions*that include ordered messages in key-value pairs [23]. Apache Samza provides flexible deployment options (run anywhere), processing and transforming data from any source, and it can be added as a client library in Java/Scala applications. Apache Samza relies on Apache Kafka, which is employed to develop stateful applications to process data in real time, providing continuous computation [58].

C. STORAGE TECHNOLOGIES

This section presents storage technologies of big data value chain and the presented technologies are included based on cited research works and public comparisons of technologies provided by DB-ENGINES [9]. Storage technologies are presented under two categories that are temporary and

TABLE 2. Comparison of acquisition technologies.

Features	Spark	Storm	Kafka	Flink	Samza
Batch Support	Yes	No	Yes	Yes	No
Latency	High	Low	Very Low	Low	High
Ordering	DStream ordering	Not guaranteed	Partition order	Not guaranteed	Partition order
Processing Model	micro-batching	micro-batching	event-at-a-time	event-at-a-time	event-at-a-time
Processing Guarantees	exactly-once	at-least-once, exactly-once	exactly-once	exactly-once	at-least-once
PL Support	Scala, Python, Java, R, C#, F#	Use with any language	Java, C/C++, Go, .NET, Python, Ruby	Scala, Python, Java, SQL	Java, Scala
Recovery	self-recovery	Checkpoint	Checkpoint	Checkpoint	Checkpoint
State Management	stateless	in-memory state	Local state	stateful	stateless stateful
Written in	Scala	Clojure, Java	Scala, Java	Java, Scala	Scala, Java
Watermark	Yes	Yes	No	Yes	No
Quality Attributes	Fault tolerance, Scalability, Reliability, High Throughput, No data loss (Durability)				

TABLE 3. Comparison of in-memory technologies.

Features	Memcached	Hazelcast	Redis	Ignite	Riak
Consistency	Eventual	Immediate, Eventual	Eventual	Immediate	Eventual
Durability	No	Yes	Yes	Yes	Yes
Partitioning	None	Sharding	Sharding	Sharding	Sharding
Released at	2003	2008	2009	2015	2009
Supported data types	String, Objects	SQL types are supported	Strings, hashes, lists, sets, sorted sets	Binary objects	No predefined data types
SQL support	No	SQL-like querying	No	ANSI-99	No
Stored Procedures	No	Event Listener, Executor services	Yes	Compute grid and cache interceptors	JavaScript, Erlang
Triggers	No	Events	Yes	cache interceptors and events	Pre-commit and post-commit hooks
Written in	C	Java	C	C++, Java, .NET	Erlang

permanent. Each category and related technologies are explained next.

1) TEMPORARY STORAGE TECHNOLOGIES

Temporary storage technologies are considered in-memory storage systems and queuing technologies (see Fig. 1). These technologies play a crucial role in handling the velocity challenge of big data by temporarily storing data and supporting real-time analytics.

In-memory stores (also called NoSQL key-value stores) use a key-value model to create data collections in key-value pairs [59]. Keys act like indexes [60] to retrieve associated values that can be any supported data type (string, list, set, hash) by applied in-memory technology [61], [62]. The distinctive characteristics of in-memory data stores are efficiently using disk and Random Access Memory (RAM) of the underlying system via allocating a pre-determined portion of RAM to keep temporary or frequently accessed data to provide fast answers on demand. For example, keeping temporary session information that is not needed to be stored permanently or managing customers' shopping cart/purchase activities on e-commerce platforms are handled by in-memory data stores [63]. According to DB-ENGINES [9], the top five open-source in-memory data storage systems out of 56 technologies are Redis, Memcached, Hazelcast, Ignite, and Riak. A technical comparison is provided in Table 3.

Queuing technologies are developed to handle the velocity of big data by allowing multiple clients concurrently to insert a large number of requests into queues and retrieve requests (or any queued data) from queues based on the *First in First Out* (FIFO) convention [64]. Moreover, queuing technologies enable partitioning, replication, and ordering of acquired data from various sources, supporting multiple programming languages to create reliable, durable (not losing data), and persistent message queues [65]. In addition, queuing technologies serve as a buffer between data sources and consumer applications that manage backpressure from slower downstream components to create a convenient environment for data processing [66]. IBM MQ, Apache ActiveMQ, RabbitMQ, ZeroMQ, and KubeMQ are open-source queuing technologies utilized in streaming processing, and Table 4 provides a fair technical comparison of these technologies.

2) PERMENANT STORAGE TECHNOLOGIES

In this big data era, dealing with large volumes of various types of structured, unstructured, and semi-structured data is a challenging duty. Properly storing, modeling, managing, and querying large amounts of data is crucial. Due to the importance of adequately storing data, hundreds of database

TABLE 4. Comparison of open-source queuing technologies.

Features	IBM MQ	ActiveMQ	RabbitMQ	ZeroMQ	KubeMQ
Architecture	Service-oriented	Master-slave	Master-slave	Pub/Sub	Message-based
Comm. Protocols	MQTT	AMQP, STOMP MQTT	AMQP	RFC 23, ZMTP	gRPC, TLS
Delivery Mode	One-time	One-time	One-time	Deliver all parts or none	At-least-once, at-most-once
Persistence	Memory Disk	Memory Disk	Memory Disk	Memory	Memory, Disk
Quality Features	Flexibility Reusability Availability	Reliability Scalability	Reliability Availability	Scalability Efficiency	Flexibility Scalability Performance
Released at	1993	2004	2007	2007	2017
Written in	Not Available	Java	Erlang	C++	Go

TABLE 5. Comparison of Rdbms technologies.

Features	Oracle	IBM Db2	MySQL	MSSQL	PostgreSQL
Complex Data Types	No support	No support	No support	No support	Hstore, Array JSON, UUID, XML
Partitioning	Sharding, Horizontal partitioning	Sharding	Sharding, Horizontal partitioning	Sharding,Horizontal partitioning	Partitioning by range, list, hash
Page Size	4 KB	4 KB	16 KB	8 KB	8 KB
Released at	1980	1983	1995	1989	1989
Stored Procedures	PL/SQL	SQL	Propriety syntax	Transact SQL	UDFs via PL/pgSQL
Written in	C, C++	C, C++	C, C++	C++	C

management systems (DBMS) have been developed [9] to handle storage-oriented challenges to accomplish numerous user demands. DBMS play a critical role in providing concurrent accessibility and security, allowing backup and crash recovery, enabling data modeling capability and data independence, minimizing data redundancy, enhancing data integrity and consistency, increasing end-user productivity, and allowing efficient data management [67]. This section presents distinctive characteristics of RDBMS, NoSQL, and NewSQL permanent storage technologies.

Since the 1980s, RDBMS became available for commercial use [68], and today more than one hundred and sixty RDBMS are actively used in various domains [9]. RDBMS was developed based on ACID (Atomicity, Consistency, Isolation, Durability) properties, making RDBMS more suitable for storing atomic, structured, and not complex data. These features are critical for banking, financial and enterprise applications to accomplish immediate consistency, speed, security, and integrity. According to DB-ENGINES [9], the top five RDBMS out of 155 technologies are Oracle, MySQL, Microsoft SQL Server (MSSQL), PostgreSQL, and IBM Db2. Feature comparison of these RDBMS is provided in Table 5. These five RDMBS share features supported by all that are triggers, immediate consistency, ACID, concurrency, durability, SQL querying support, foreign keys (referential integrity), access methods (ODBC, JDBC), indexing mechanism is B-Tree, and primitive data types (number, date, and string).

On the other hand, ACID characteristics of RDBMS force centralized databases and do not enable replications of data in a distributed fashion. Also, RDBMS provides little or no support for storing large, complex, and diverse semi-structured and unstructured big data. In addition, scalability support of RDBMS is performed by scale up (vertical) that requires changing existing hardware with vendor-dependent expensive hardware. Therefore, to eliminate presented restrictions of RDBMS and to handle the challenges of big data, NoSQL data stores have emerged.

NoSQL stands for "*Not Only SQL*," which explains there exists a flexible schema but not restricted as RDBMS's schema. Milestone development in NoSQL has been triggered by the development of Google's Big Table [69] and Amazon's DynamoDB [70], [71]. NoSQL data stores provide capabilities such as storing large amounts of big data in different formats under a flexible and distributed schema, achieving horizontal scalability on commodity hardware, distributing copies of data across machines to increase availability and performance, and eliminating a single point of failure [72]. NoSQL data storage systems have been developed based on BASE (Basically Available, Soft State, and Eventual Consistency) features to relax the restriction of ACID properties. NoSQL data storage systems are presented under the following categories that are key-value, document, wide-column, and graph [73], [74], [75], [76]. Popular *key-value stores* are presented in Section V/C/1.

Document stores allow storing semi-structured data. A document is identified with a unique identifier and can contain an arbitrary number of key-value pairs in any nested form without schema restriction. Documents can be persisted in BSON, JSON, and XML formats [77]. A document

TABLE 6. Comparison of NoSQL document storage technologies.

Features	CouchDB	MongoDB	Couchbase	Databricks	Realm
Consistency	Eventual	Eventual Immediate	Eventual Immediate	Immediate	Immediate
Cloud based only	No	No	No	Yes	No
Device Support	Mobile Desktop	Mobile Desktop	Mobile Desktop Embedded	Cloud Support	Smartphone Tablet Desktop
Full Text Search	No	Lucene based	No	Using Elasticsearch	Yes
Quality Features	Reliability Scalability	Scalability Availability	Versatility Performance Scalability	Performance	Scalability Performance
Released at	2005	2009	2011	2013	2014
Stored Procedures	View functions	JavaScript	Functions and timers	UDFs and aggregates	No
Triggers	Yes	Yes	TAP protocol	Yes	Change Listener
Written in	Erlang	C++	C++, Erlang	Based on Apache Spark	C++

TABLE 7. Comparison of NoSQL wide-column storage technologies.

Features	Cassandra	HBase	Accumulo	ScyllaDB
Consistency	Eventual Immediate	Eventual Immediate	Eventual	Eventual Tunable
Map Reduce	Yes	Yes	Yes	No
Querying mechanism	CQL	Apache Drill	Relies on HDFS	CQL
Quality Features	Scalability Availability Fault-tolerance Reliability	Scalability	Scalability Fault-tolerance Performance	Scalability Availability Performance
Released at	2008	2008	2008	2015
Replication	Replication factor	Multi-source	Automatic replication	Replication factor
Stored procedures	No	Yes	No	Yes
Triggers	Yes	Yes	Yes	No
Written in	Java	Java	Java	C++

store can store any number of collections regarding available disk size. According to DB-ENGINES [9], MongoDB, Databricks, Couchbase, CouchDB, and Realm are the top five open-source NoSQL document stores out of 47 technologies, and Table 6 provides a comparison of these data stores.

Wide-column (columnar or extensible record stores) [60] stores are inspired by Google's Big Table and aim to provide a flexible schema to store complex semi-structured or unstructured large amounts of data in various formats [78]. Wide-column stores provide column families like flexible tables but unlike RDBMS tables. A column family enables the creation of row keys and associated columns. Each row key is associated with many numbers of columns (keys) that contains data (value). The columns associated with a particular row key are stored together on a disk [79].

Moreover, data replication is performed based on row keys on multiple nodes. According to [9], Cassandra, HBase, Accumulo, and ScyllaDB are popular wide-column data stores out of 13 wide-column stores. Table 7 provides a feature comparison of these data stores. In addition, shared features of presented data stores are open-source, access protocols (thrift), distributed nodes, concurrency, durability, and no support for foreign keys.

Graph data stores are another important NoSQL category that enables storing real-world entities and keeping relations among these entities via a schema-free graph model. A graph model includes a network of nodes representing real-world entities and directed edges for relations and properties to keep entity features in key-value pairs [80], [81]. According to DB-ENGINES [9], the top five open-source out of 37 Graph NoSQL data stores are Neo4j, Virtuoso, ArangoDB, OrientDB, and JanusGraph. Comparing these storage technologies are presented in Table 8 using distinctive features. The shared features for these graph stores are fault tolerance, concurrency, performance, durability, not supporting Map Reduce, graph data model support, and supporting ACID [73]. Another essential data store category is NewSQL [70].

NewSQL databases are considered under RDBMS, keeping ACID features of RDBMS and adding the flexibility of NoSQL data stores. Popular open-source NewSQL databases are VoltDB, NuoDB, and SingleStore [40], [82] and Table 9 compares these technologies. NewSQL databases provide full ACID support, querying via SQL, horizontal scaling, durability, concurrency, and high availability.

D. ANALYTICS TECHNOLOGIES

In Section V/B, five open-source streaming analytics tools and feature comparisons are provided (see Table 2). In this section, comparison of five open-source batch data processing technologies that are Spark [52], Pig [83], Hive [84], Drill [85], and Impala [86] presented.

TABLE 8. Comparison of NoSQL graph storage technologies.

Features	Virtuoso	Neo4j	OrientDB	ArangoDB	JanusGraph
Consistency	Immediate	Casual Eventual Immediate	MVCC	Eventual Immediate	Eventual Immediate
Partitioning	Yes	No	Sharding	Sharding	Relies on storage backend
Query Language	SPARQL	Cypher	SQL Like	AQL	Gremlin
Quality Features	Scalability	Scalability Availability Reliability	Reliability Flexibility	Flexibility Scalability	Scalability
Released	1998	2007	2010	2012	2017
Stored Procedures	Virtuoso PL	UDFs and functions	Java JavaScript	JavaScript	Yes
Triggers	Yes	Event Handler	Hooks	No	Yes
Written in	C	Java Scala	Java	C++	Java

TABLE 9. Comparison of NewSQL storage technologies.

Features	VoltDB	SingleStore	NuoDB
Consistency	Transactional	Immediate	Immediate
Partitioning	Sharding	Hash partitioning	Dynamic partition
Replication	Multi-source Source-replica	Source-replica	On-demand replication
Released at	2010	2013	2013
Quality Features	Scalability Availability Reliability Performance	Performance Fault-tolerance Simplicity	Scalability Availability Flexibility
Stored Procedures	Java	UDFs	Java, SQL
Triggers	No	No	Yes
Written in	Java, C++	C++, Go	C++

TABLE 10. Comparison of selected batch processing technologies.

Features	Pig	Spark	Hive	Drill	Impala
Analytics support	Exploratory HDFS data analytics	Machine learning, graph processing	Process structured large datasets	Automatic	Fast data processing
Optimize	Automatic	Lazy evaluation	Cost based	Caching based	Automatic
Processing Model	Map- reduce	Micro-batches	Map-reduce	Drillbit	MPP (Massively Parallel Processing)
Querying Language	Pig Latin	Spark SQL	HiveQL	ANSI SQL	HiveQL
Quality Attributes	Extensibility Flexibility	Reusability Scalability Flexibility	Scalability	Performance Flexibility Extensibility	Scalability
Recovery	code recovery	Self-recovery	Hive DR	Recovery beforehand	No recovery
Released at	2008	2009	2010	2014	2017
Supported data types	All types of data	Parquet, Text JSON, ORC, CSV, Avro,	Parquet, ORC, CSV/TSV	Parquet, JSON, Text	Parquet, Avro, Text, RCFile, Sequence file

Table 10 provides a comparison of these technologies. Shared features for these technologies are Apache technologies, ETL (Extract Transform Load) support, and UDF (User-defined Functions) support.

VI. RELATED WORKS

In Table 11, a comparison of related works is provided regarding technologies utilized in the acquisition, storage, and analytics stages of the big data value chain. Unlike other related works, this paper presents big data technologies employed in all stages of acquisition, storage and analytics.

In [7], the concept of big data is explained in terms of the big data value chain. For each stage of the big data value chain, related terminology and features are presented, as are big data analytics methods, systems, and benchmarks. In [19], [43], and [53], an overview of big data, big data-related technologies, the big data value chain, challenges, examples of big data applications, the Hadoop ecosystem, and open research questions are presented. In [23], a comprehensive survey on big data concepts, systems, and technologies is provided in MapReduce, NoSQL, ML tools, Hadoop technologies, and data processing and querying tools.

TABLE 11. Comparison of related works regarding big data value chain technologies.

Year	Ref.	**Acquisition**	**Temporary Storage**		**Permanent Storage**			**Analytics**	
		Stream Processing	In-memory	Queuing	RDBMS	NoSQL	NewSQL	Streaming	Batch
2014	[7]	✓	✓	-	-	✓	-	✓	✓
	[19]	-	-	-	✓	✓	-	-	✓
2016	[53]	✓	-	-	-	✓	-	✓	✓
2017	[43]	✓	-	-	✓	✓	-	✓	✓
2018	[23]	✓	✓	✓	-	✓	-	✓	✓
2019	[29]	-	✓	-	-	✓	-	✓	✓
	[37]	✓	-	-	-	✓	-	✓	✓
2020	[40]	-	✓	-	✓	✓	✓	-	-
	[21]	✓	-	-	-	-	-	✓	✓
2022	[46]	✓	-	-	-	✓	-	✓	✓
	[87]	✓	✓	-	-	✓	-	✓	-
2023	*This work*	✓	✓	✓	✓	✓	✓	✓	✓

The meaning of '✓' is the reference presents a technology from given category and '-' indicates it does not contain a technology in the given category

In [29], a CAP-based classification of data storage systems is provided, and NoSQL data storage systems utilized in streaming data processing are presented. In [37], big data characteristics, data processing types, Hadoop clusters, challenges and success factors, applications in big data analytics, and trends are presented. In [40], timelines, characteristics, sources, and definitions of big data are provided. Also, a comparison of data storage technologies, the CAP theorem, and the grand challenges of big data are presented. In [21], the definitions, characteristics, and challenges of big data are presented. A classification for big data, domains and data sources, batch processing, stream processing, and big data analytics techniques are provided.

In [46], recent big data analytics tools and their features are presented. Data-driven industrial applications and the challenges of big data analytics projects are discussed. The strengths and weaknesses of the presented big data technologies are discussed. In [87], big data processing and storage systems utilized in real-time data processing are presented. Popular real-time data processing technologies and NoSQL data storage systems used in real-time data processing is provided.

Furthermore, beyond the presented related works, this paper aims to provide a technology-oriented perspective for all stages (see Table 11) of big data value chain to support developers of data-intensive systems to choose the right technology in their jobs for a demanded stage of big data value chain.

VII. APPLICATIONS AND USE CASES

Data is everywhere, and data processing and analytics are crucial to accomplishing various purposes in almost all diverse domains, such as education, transportation, healthcare, business, crisis informatics, etc. [46]. For example, crisis informatics researchers in academia study crisis data to explore public behaviors before, during, and after emergencies and to investigate disaster-affected people during emergencies to help and support crisis management [88].

Although each domain has its own requirements, priorities, and purposes for performing data processing and analytics tasks, big data technologies are utilized in diverse domains to handle various challenges and perform demanded actions. Thus, this section includes example use cases listed in Table 12 from different domains, focusing on the stages of the big data value chain and related big data technologies. These research works were selected during the search through the terms mentioned in the related works section by focusing on their relevance, recent publication, and technology use in the stages of the big data value chain.

Table 12 presents a list of published research works in emergency management, education, healthcare, construction, and IoT for big data analytics. In addition, Table 12 provides published application examples and insights for big data technology use in stages of the big data value chain for big data researchers to support their own research efforts and developers to support their building tasks of data-intensive systems.

Lastly, the technology selection can be based on requirements, priorities, user demands, and domain needs. Therefore, one size doesn't fit all; for example, one technology can be used in different domains, such as Kafka or Spark, in the acquisition stage. While one system can only utilize one data storage system (MongoDB [90] or Cassandra [92]), another system (reference [96]) requires the inclusion of multiple and diverse data storage systems. Thus, one of the main concerns is accomplishing demanded functionality and meeting requirements while developing data-intensive systems.

VIII. DISCUSSION AND LIMITATIONS

In this big data era, various domains, diverse requirements, various needs, evolving hardware devices, and constantly changing software tools and environments create a variety of challenges for developers of data-intensive systems [97]. Each one of these aspects triggers the development of various tools and technologies to fulfill user needs, handle a variety of challenges, and accomplish domain requirements.

Moreover, developers of data-intensive systems can learn from the best examples of the usage of big data technologies in diverse domains. Although each technology in the big data value chain fulfills its duty in a data-intensive system, from

TABLE 12. Use cases regarding big data value chain technologies.

Reference	Purpose & Domain	Technologies of Big Data Value Chain		
		Acquisition	Storage	Analytics
[89]	Waste analytics & Construction	Flume	Hadoop Neo4j	Spark
[90]	OpenStreetMap & Crisis Informatics	Spark	MongoDB	Spark
[91]	Real-time tweet collection and analytics & Crisis Informatics	Spark	Redis, RabbitMQ, Cassandra	Spark
[92]	Real-time tweet analytics & Crisis Informatics	Kafka	Cassandra	Spark
[93]	Learning analytics & Education	ETL	Hadoop, SQL Server	Hive
[94]	IoT & Transportation	From Sensors to Cloud	Hadoop, HBase, MySQL	Pig
[95]	Social big data analytics & Social Media Analysis	Kafka	Cassandra	Spark
[96]	Big data analytics & Healthcare	Kafka	MongoDB, Cassandra, PostgreSQL, Redis, Hadoop	Spark, Hive, Drill

the developing angle, developers must incorporate various tools, utilize different programming languages, and come up with feasible techniques to accomplish harmony in the demanding system. Thus, carefully selecting the right combination of big data technology is paramount. At this point, this study supports developers of data-intensive systems in their technology selection based on the presented technology features and comparisons.

On the other hand, this paper does not include an example of a specific application; however, it aims to share the cumulatively gained results of previously developed data-intensive systems [78], [79], [91], [98], [99]. Therefore, the following suggestions for researchers and developers of data-intensive systems are paramount: having domain knowledge to understand user needs; building a developer team that is eager to learn, open for collaboration, and diligent in solving unexpected issues; utilizing the right set of cutting-edge technologies for acquisition, storage, and analytics; and developing feasible techniques for analyzing data to accomplish a variety of purposes in different domains [78].

Additionally, there are potential limitations associated with this paper. First, it is possible that additional studies exist that were not included in this work because they did not mention the big data value chain, did not focus on big data technologies, or were not discovered through Web of Science searches. There may be relevant documents that are not publicly accessible or written in a language other than English or Turkish. In addition, there are numerous papers on big data analytics; however, in Section VII, the presented use cases include the utilization of big data technology at all phases of the big data value chain.

IX. CONCLUSION

In this paper, a comparative perspective on big data technologies is provided from the presented big data value chain model. To conclude, first, in Section II, the concept of big data is presented by providing definitions, characteristics, big data processing paradigms, and data analytics types; second, a comparative perspective on technologies in each stage of the big data value chain model is provided, and their usage purposes and capabilities are presented. Thirdly, a selection of applications and use cases for big data implementations in various domains is presented. Finally, this study aims to assist big data researchers and developers of data-intensive systems by providing an overview of big data processing concepts, comparing open-source technologies used in the acquisition, storage, and analytics stages, and presenting example use cases from valuable research works. In addition, this paper also illustrates technology usage throughout the phases of the big data value chain in order to give insight for future data-intensive system development.

REFERENCES

[1] Domo Company. *Data Never Sleeps 9.0*. Accessed: Feb. 14, 2023. [Online]. Available: https://www.domo.com/learn/infographic/data-never-sleeps-9

[2] S. Sakr, "Big data processing stacks," *IT Prof.*, vol. 19, no. 1, pp. 34–41, Jan. 2017, doi: 10.1109/MITP.2017.6.

[3] A. Oussous, F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big data technologies: A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 4, pp. 431–448, Oct. 2018, doi: 10.1016/j.jksuci.2017.06.001.

[4] D. Demirol, R. Das, and D. Hanbay, "Büyük veri üzerine perspektif bir bakis," in *Proc. Int. Artif. Intell. Data Process. Symp. (IDAP)*, Sep. 2019, pp. 1–9, doi: 10.1109/IDAP.2019.8875902.

[5] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. M. Ali, M. Alam, M. Shiraz, and A. Gani, "Big data: Survey, technologies, opportunities, and challenges," *Sci. World J.*, vol. 2014, pp. 1–18, Jul. 2014, doi: 10.1155/2014/712826.

[6] I. Yaqoob, I. A. T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N. B. Anuar, and A. V. Vasilakos, "Big data: From beginning to future," *Int. J. Inf. Manag.*, vol. 36, no. 6, pp. 1231–1247, Dec. 2016, doi: 10.1016/j.ijinfomgt.2016.07.009.

[7] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014, doi: 10.1109/ACCESS.2014.2332453.

[8] L. Zhu, L. Bass, and X. Xu, "Data management requirements for a knowledge discovery platform," in *Proc. WICSA/ECSA Companion Volume*, Aug. 2012, pp. 169–172, doi: 10.1145/2361999.2362036.

[9] Solid-IT. *DB-Engines*. Accessed: Feb. 14, 2023. [Online]. Available: https://db-engines.com/en/

[10] M. Cox and D. Ellsworth, "Application-controlled demand paging for out-of-core visualization," in *Proc. Visualizat.*, 1997, pp. 235–244, doi: 10.1109/visual.1997.663888.

[11] 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, doi: 10.1007/s11227-015-1501-1.

[12] J. Manyika, M. C. Brown, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition and productivity," McKinsey Global Inst., San Francisco, CA, USA, Tech. Rep., 2011.

[13] J. Gantz and D. Reinsel. (Jan. 2011). *Extracting Value From Chaos*. [Online]. Available: www.idc.com

[14] S. Madden, "From databases to big data," *IEEE Internet Comput.*, vol. 16, no. 3, pp. 4–6, May/Jun. 2012, doi: 10.1109/MIC.2012.50.

[15] M. Cooper and P. Mell. *Tackling Big Data Complexity*. Accessed: Feb. 14, 2023. [Online]. Available: https://bigdata.nist.gov/_uploadfiles/M0065_v1_4451775754.pdf

[16] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 42–47, doi: 10.1109/CTS.2013.6567202.

[17] *NIST Special Publication 1500-1—NIST Big Data Interoperability Framework: Volume 1, Definitions*, NIST Special, Gaithersburg, MD, USA, 2015, doi: 10.6028/NIST.SP.1500-1.

[18] Gartner. *Gartner*. Accessed: Feb. 14, 2023. [Online]. Available: https://www.gartner.com/en/information-technology/glossary/big-data

[19] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014, doi: 10.1007/s11036-013-0489-0.

[20] M. Ianni, E. Masciari, and G. Sperlí, "A survey of big data dimensions vs social networks analysis," *J. Intell. Inf. Syst.*, vol. 57, no. 1, pp. 73–100, Aug. 2021, doi: 10.1007/s10844-020-00629-2.

[21] A. Mohamed, M. K. Najafabadi, Y. B. Wah, E. A. K. Zaman, and R. Maskat, "The state of the art and taxonomy of big data analytics: View from new big data framework," *Artif. Intell. Rev.*, vol. 53, pp. 989–1037, Feb. 2020, doi: 10.1007/s10462-019-09685-9.

[22] C. Kacfah Emani, N. Cullot, and C. Nicolle, "Understandable big data: A survey," *Comput. Sci. Rev.*, vol. 17, pp. 70–81, Aug. 2015, doi: 10.1016/j.cosrev.2015.05.002.

[23] T. R. Rao, P. Mitra, R. Bhatt, and A. Goswami, "The big data system, components, tools, and technologies: A survey," *Knowl. Inf. Syst.*, vol. 60, pp. 1165–1245, Sep. 2018, doi: 10.1007/s10115-018-1248-0.

[24] M. K. Saggi and S. Jain, "A survey towards an integration of big data analytics to big insights for value-creation," *Inf. Process. Manag.*, vol. 54, no. 5, pp. 758–790, Sep. 2018, doi: 10.1016/j.ipm.2018.01.010.

[25] M. D. A. Praveena and B. Bharathi, "A survey paper on big data analytics," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2017, pp. 1–9, doi: 10.1109/ICICES.2017.8070723.

[26] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger, "Social media analytics—Challenges in topic discovery, data collection, and data preparation," *Int. J. Inf. Manag.*, vol. 39, pp. 156–168, Apr. 2018, doi: 10.1016/j.ijinfomgt.2017.12.002.

[27] D. Staegemann, M. Volk, A. Saxena, M. Pohl, A. Nahhas, R. Häusler, M. Abdallah, S. Bosse, N. Jamous, and K. Turowski, "Challenges in data acquisition and management in big data environments," in *Proc. 6th Int. Conf. Internet Things, Big Data Secur.*, 2021, pp. 193–204, doi: 10.5220/0010429001930204.

[28] R. Casado and M. Younas, "Emerging trends and technologies in big data processing," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 8, pp. 2078–2091, Jun. 2015, doi: 10.1002/cpe.3398.

[29] T. B. Doguç and A. A. Aydin, "CAP-based examination of popular NoSQL database technologies in streaming data processing," in *Proc. Int. Artif. Intell. Data Process. Symp. (IDAP)*, Sep. 2019, pp. 1–6, doi: 10.1109/IDAP.2019.8875874.

[30] R. L. D. C. Costa, J. Moreira, P. Pintor, V. dos Santos, and S. Lifschitz, "A survey on data-driven performance tuning for big data analytics platforms," *Big Data Res.*, vol. 25, Jul. 2021, Art. no. 100206, doi: 10.1016/j.bdr.2021.100206.

[31] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008, doi: 10.1145/1327452.1327492.

[32] S. N. Khezr and N. J. Navimipour, "MapReduce and its applications, challenges, and architecture: A comprehensive review and directions for future research," *J. Grid Comput.*, vol. 15, no. 3, pp. 295–321, Sep. 2017, doi: 10.1007/s10723-017-9408-0.

[33] A. Kejariwal, S. Kulkarni, and K. Ramasamy, "Real time analytics: Algorithms and systems," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 2040–2041, Aug. 2015, doi: 10.14778/2824032.2824132.

[34] G. B. Kumar, "An encyclopedic overview of 'big data' analytics," *Int. J. Appl. Eng. Res.*, vol. 10, no. 3, pp. 5681–5705, 2015.

[35] N. A. Ghani, S. Hamid, I. A. Targio Hashem, and E. Ahmed, "Social media big data analytics: A survey," *Comput. Hum. Behav.*, vol. 101, pp. 417–428, Dec. 2019, doi: 10.1016/j.chb.2018.08.039.

[36] I. Ahmed, M. Ahmad, G. Jeon, and F. Piccialli, "A framework for pandemic prediction using big data analytics," *Big Data Res.*, vol. 25, Jul. 2021, Art. no. 100190, doi: 10.1016/j.bdr.2021.100190.

[37] I. A. Ajah and H. F. Nweke, "Big data and business analytics: Trends, platforms, success factors and applications," *Big Data Cognit. Comput.*, vol. 3, no. 2, p. 32, Jun. 2019, doi: 10.3390/bdcc3020032.

[38] E. Sirin and H. Karacan, "A review on business intelligence and big data," *Int. J. Intell. Syst. Appl. Eng.*, vol. 5, no. 4, pp. 206–215, 2017.

[39] M. R. Ahmed, M. A. Khatun, M. A. Ali, and K. Sundaraj, "A literature review on NoSQL database for big data processing," *Int. J. Eng. Technol.*, vol. 7, no. 2, p. 902, Jun. 2018, doi: 10.14419/ijet.v7i2.12113.

[40] Z. Lashkaripour, "The era of big data: A thorough inspection in the building blocks of future generation data management," *Int. J. Sci. Technol. Res.*, vol. 9, no. 10, pp. 321–330, 2020.

[41] M. F. Khan, M. Azam, M. A. Khan, F. Algarni, M. Ashfaq, I. Ahmad, and I. Ullah, "A review of big data resource management: Using smart grid systems as a case study," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–18, Oct. 2021, doi: 10.1155/2021/3740476.

[42] M. Barlow, *Real-Time Big Data Analytics: Emerging Architecture*. Sebastopol, CA, USA: O'Reilly Media, 2013, doi: 10.1007/s13398-014-0173-7.2.

[43] K. Venkatram and M. A. Geetha, "Review on big data & analytics—Concepts, philosophy, process and applications," *Cybern. Inf. Technol.*, vol. 17, no. 2, pp. 3–27, Jun. 2017, doi: 10.1515/cait-2017-0013.

[44] S. Dessureault, "Understanding big data," *CIM Mag.*, vol. 11, no. 1. p. 33, 2016.

[45] A. Z. Abualkishik, "Hadoop and big data challenges," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 12, pp. 3488–3500, 2019.

[46] A. C. Ikegwu, H. F. Nweke, C. V. Anikwe, U. R. Alo, and O. R. Okonkwo, "Big data analytics for data-driven industry: A review of data sources, tools, challenges, solutions, and research directions," *Cluster Comput.*, vol. 25, no. 5, pp. 3343–3387, 2022, doi: 10.1007/s10586-022-03568-5.

[47] Apache Software Foundation. *Hadoop*. Accessed: Feb. 14, 2023. [Online]. Available: https://hadoop.apache.org/

[48] D. Bhattacharya and M. Mitra, "Analytics on big fast data using a real time stream data processing architecture," EMC2, Hunterdon County, NJ, USA, 2013.

[49] W. Wingerath, F. Gessert, S. Friedrich, and N. Ritter, "Real-time stream processing for big data," *Inf. Technol.*, vol. 58, no. 4, pp. 186–194, Aug. 2016, doi: 10.1515/itit-2016-0002.

[50] E. E. Drakonaki and G. M. Allen, "Magnetic resonance imaging, ultrasound and real-time ultrasound elastography of the thigh muscles in congenital muscle dystrophy," *Skeletal Radiol.*, vol. 39, no. 4, pp. 391–396, Apr. 2010, doi: 10.1007/s00256-009-0861-0.

[51] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[52] Apache Software Foundation. *Spark*. Accessed: Feb. 14, 2023. [Online]. Available: https://spark.apache.org/

[53] F. Bajaber, R. Elshawi, O. Batarfi, A. Altalhi, A. Barnawi, and S. Sakr, "Big data 2.0 processing systems: Taxonomy and open challenges," *J. Grid Comput.*, vol. 14, no. 3, pp. 379–405, Sep. 2016, doi: 10.1007/s10723-016-9371-1.

[54] Apache Software Foundation. (2023). *Storm*. Accessed: Feb. 13, 2023. [Online]. Available: https://storm.apache.org/

[55] Apache Software Foundation. *Kafka*. Accessed: Feb. 14, 2023. [Online]. Available: https://kafka.apache.org/

[56] F. Bajaber, S. Sakr, O. Batarfi, A. Altalhi, and A. Barnawi, "Benchmarking big data systems: A survey," *Comput. Commun.*, vol. 149, pp. 241–251, Jan. 2020, doi: 10.1016/j.comcom.2019.10.002.

[57] Apache Software Foundation. *Flink*. Accessed: Feb. 14, 2023. [Online]. Available: https://flink.apache.org/

[58] Apache Software Foundation. *Samza*. Accessed: Feb. 14, 2023. [Online]. Available: https://samza.apache.org/

[59] A. Ribeiro, A. Silva, and A. R. da Silva, "Data modeling and data analytics: A survey from a big data perspective," *J. Softw. Eng. Appl.*, vol. 8, no. 12, pp. 617–634, 2015, doi: 10.4236/jsea.2015.812058.

[60] V. Manoj, "Comparative study of NoSQL document, column store databases and evaluation of Cassandra," *Int. J. Database Manag. Syst.*, vol. 6, no. 4, pp. 11–26, Aug. 2014, doi: 10.5121/ijdms.2014.6402.

[61] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL database: New era of databases for big data analytics—Classification, characteristics and comparison," *Int. J. Database Theory Appl.*, vol. 6, no. 4, pp. 1–14, 2013, doi: 10.1016/S0262-4079(12)63205-9.

[62] S. Ramzan, I. S. Bajwa, and R. Kazmi, "Challenges in NoSQL-based distributed data storage: A systematic literature review," *Electronics*, vol. 8, no. 5, p. 488, Apr. 2019, doi: 10.3390/electronics8050488.

[63] K. Srivastava and N. Shekokar, "A polyglot persistence approach for e-commerce business model," in *Proc. Int. Conf. Inf. Sci. (ICIS)*, Aug. 2016, pp. 7–11, doi: 10.1109/INFOSCI.2016.7845291.

[64] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Shelter Island, NY, USA: Manning Publications, 2015.

[65] G. Fu, Y. Zhang, and G. Yu, "A fair comparison of message queuing systems," *IEEE Access*, vol. 9, pp. 421–432, 2021, doi: 10.1109/ACCESS.2020.3046503.

[66] R. Betts and J. Hugg, *Fast Data: Smart and at Scale*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[67] C. Coronel, S. Morris, and P. Rob, *Database Systems: Design, Implementation, and Management*, 9th ed. Boston, MA, USA: Cengage Learning, 2011.

[68] R. Elmasri and S. B. Navathe, *Fundementals of Database Systems*, 6th ed. Reading, MA, USA: Addison-Wesley, 2010.

[69] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 1–26, Jun. 2008, doi: 10.1145/1365815.1365816.

[70] R. Cattell, "Scalable SQL and NoSQL data stores," *ACM SIGMOD Rec.*, vol. 39, no. 4, pp. 12–27, May 2011, doi: 10.1145/1978915.1978919.

[71] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, and J. Bernardino, "Choosing the right NoSQL database for the job: A quality attribute evaluation," *J. Big Data*, vol. 2, no. 1, p. 18, Dec. 2015, doi: 10.1186/s40537-015-0025-0.

[72] A. Schram and K. M. Anderson, "MySQL to NoSQL: Data modeling challenges in supporting scalability," in *Proc. 3rd Annu. Conf. Syst., Program., Appl., Softw. Humanity*, Oct. 2012, pp. 191–202, doi: 10.1145/2384716.2384773.

[73] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "NoSQL databases for big data," *Int. J. Big Data Intell.*, vol. 4, no. 3, pp. 171–185, 2017.

[74] W. Ali, M. U. Shafique, M. A. Majeed, and A. Raza, "Comparison between SQL and NoSQL databases and their relationship with big data analytics," *Asian J. Res. Comput. Sci.*, pp. 1–10, Oct. 2019, doi: 10.9734/AJRCOS/2019/v4i230108.

[75] H. Vera-Olivera, R. Guo, R. C. Huacarpuma, A. P. B. Da Silva, A. M. Mariano, and M. Holanda, "Data modeling and NoSQL databases—A systematic mapping review," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–26, Jul. 2022, doi: 10.1145/3457608.

[76] S. Sakr and A. Elgammal, "Towards a comprehensive data analytics framework for smart healthcare services," *Big Data Res.*, vol. 4, pp. 44–58, Jun. 2016, doi: 10.1016/j.bdr.2016.05.002.

[77] H. Hashem and D. Ranc, "Pre-processing and modeling tools for big-data," *Found. Comput. Decis. Sci.*, vol. 4, no. 52, pp. 3–27, 2016, doi: 10.1515/fcds-2016-0009.

[78] A. A. Aydin and K. M. Anderson, "Data modelling for large-scale social media analytics: Design challenges and lessons learned," *Int. J. Data Mining, Model. Manag.*, vol. 12, no. 4, p. 386, 2020, doi: 10.1504/IJDMMM.2020.111409.

[79] A. A. Aydin, "Incremental data collection & analytics the design of next-generation crisis informatics software," Ph.D. thesis, Dept. Comput. Sci., Univ. Colorado Boulder, ProQuest Diss. Publishing, Boulder, CO, USA, 2016. [Online]. Available: https://www.proquest.com/pagepdf/1834583278/Record/9F7C2D640FDE4BCCPQ/3?accountid=16268

[80] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Comput. Surv.*, vol. 40, no. 1, pp. 1–39, Feb. 2008, doi: 10.1145/1322432.1322433.

[81] M. Besta, R. Gerstenberger, E. Peter, M. Fischer, M. Podstawski, C. Barthels, G. Alonso, and T. Hoefler, "Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries," 2019, *arXiv:1910.09017*.

[82] J. Ryan. (2019). *Big Data Velocity in Plain English*. [Online]. Available: https://www.voltdb.com/wp-content/uploads/2018/02/VoltDB_BigData_eBook_Feb2018-v2.pdf

[83] Apache Software Foundation. *Apache Pig*. Accessed: Feb. 14, 2023. [Online]. Available: https://pig.apache.org/

[84] Apache Software Foundation. *Apache Hive*. Accessed: Feb. 14, 2023. [Online]. Available: https://hive.apache.org/

[85] Apache Software Foundation. *Apache Drill*. Accessed: Feb. 14, 2023. [Online]. Available: https://drill.apache.org/

[86] Apache Software Foundation. *Impala*. Accessed: Feb. 14, 2023. [Online]. Available: https://impala.apache.org/

[87] U. Kekevi and A. A. Aydin, "Real-time big data processing and analytics: Concepts, technologies, and domains," *Comput. Sci.*, vol. 55, no. 35, pp. 1–100, Nov. 2022, doi: 10.53070/bbd.1204112.

[88] A. A. Aydin, "Prominent quality attributes of crisis software systems: A literature review," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 28, no. 5, pp. 2507–2522, Sep. 2020, doi: 10.3906/elk-1911-5.

[89] M. Bilal, L. O. Oyedele, O. O. Akinade, S. O. Ajayi, H. A. Alaka, H. A. Owolabi, J. Qadir, M. Pasha, and S. A. Bello, "Big data architecture for construction waste analytics (CWA): A conceptual framework," *J. Building Eng.*, vol. 6, pp. 144–156, Jun. 2016, doi: 10.1016/j.jobe.2016.03.002.

[90] J. Anderson, R. Soden, K. M. Anderson, M. Kogan, and L. Palen, "EPIC-OSM: A software framework for OpenStreetMap data analytics," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 5468–5477, doi: 10.1109/HICSS.2016.675.

[91] A. Aydin and K. Anderson, "Batch to real-time: Incremental data collection & analytics platform," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017, pp. 5911–5920. [Online]. Available: http://hdl.handle.net/10125/41876

[92] S. Jambi and K. M. Anderson, "Engineering scalable distributed services for real-time big data analytics," in *Proc. IEEE 3rd Int. Conf. Big Data Comput. Service Appl. (BigDataService)*, Apr. 2017, pp. 131–140, doi: 10.1109/BigDataService.2017.22.

[93] A. Klašnja-Milicević, M. Ivanović, and Z. Budimac, "Data science in education: Big data and learning analytics," *Comput. Appl. Eng. Educ.*, vol. 25, no. 6, pp. 1066–1078, Nov. 2017, doi: 10.1002/cae.21844.

[94] M. M. Hussain, M. M. S. Beg, M. S. Alam, M. Krishnamurthy, and Q. M. Ali, "Computing platforms for big data analytics in electric vehicle infrastructures," in *Proc. 4th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2018, pp. 138–143, doi: 10.1109/BIGCOM.2018.00029.

[95] B. A. Hammou, A. A. Lahcen, and S. Mouline, "Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics," *Inf. Process. Manag.*, vol. 57, no. 1, Jan. 2020, Art. no. 102122, doi: 10.1016/j.ipm.2019.102122.

[96] S. Imran, T. Mahmood, A. Morshed, and T. Sellis, "Big data analytics in healthcare—A systematic literature review and roadmap for practical implementation," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 1, pp. 1–22, Jan. 2021, doi: 10.1109/JAS.2020.1003384.

[97] F. Brooks, "No silver bullet: Essence and accident of software engineering," *IEEE Softw.*, vol. S-20, no. 4, pp. 10–19, Apr. 1987, doi: 10.1109/MC.1987.1663532.

[98] A. A. Aydin and K. M. Anderson, "Incremental sorting for large dynamic data sets," in *Proc. IEEE 1st Int. Conf. Big Data Comput. Service Appl.*, Mar. 2015, pp. 170–175, doi: 10.1109/BigDataService.2015.35.

[99] K. M. Anderson, A. A. Aydin, M. Barrenechea, A. Cardenas, M. Hakeem, and S. Jambi, "Design challenges/solutions for environments supporting the analysis of social media data in crisis informatics research," in *Proc. 48th Hawaii Int. Conf. Syst. Sci.*, Jan. 2015, pp. 163–172, doi: 10.1109/HICSS.2015.29.

AHMET ARIF AYDIN received the Ph.D. degree in computer science from the University of Colorado Boulder, in 2016, with a specialization in architectural design for data analytics platforms. He is currently an Assistant Professor with the Computer Engineering Department, Inonu University. His current research interests include software engineering, data-intensive system design, crisis informatics, big data analytics, data modeling, algorithm design for analytics, parallel processing, and machine learning.