Chapter 1

Introduction and Motivation

In the modern era humankind collects vast amounts of *information* in various areas, e.g., social networks, climate, healthcare and business just to mention some of the most prominent fields. Information, in general, refers to facts, details or knowledge about somebody/something ([8], [12]). This information can be extremely valuable to support our decision making and enhance our ability to take data-driven decisions.

In the field of IT a more technical term, *data*, is used to refer to a factual information used as a basis for reasoning or decision-making ([8], [12]). The word itself originates from the latin word "datum" ("data" is a plural form), which is the neuter perfect passive participle of the verb "dō" (I give). The word "datum" means 'given', but occasionally it can also mean a gift or a present.

Analysis refers to the detailed study or examination of something complex in order to understand its nature and essential features ([8], [12]). Based on this definition *data analysis* refers to the detailed study or examination of data in order to understand its nature and essential features to use it for decision making.

According to reports from International Data Corporation (IDC) [6], 64,2 ZB[1] of data was created or replicated in 2020, however only a small percentage of this data is stored and retained to 2021. It is estimated that data creation and replication will experience a compound annual growth rate of 23% in the 2020-2025 forecast period. The fastest growing data segment is IoT data, followed by social media. In an earlier study CISCO estimated the

[1] 1 ZB = 10^{21} B = 10^{12} GB

internet traffic to reach 235,7 EB[2] per month during 2021 [5].

To cope with this fast increase of data size new paradigms and tools evolved, collectively known as *big data*. Compared to traditional data management methods, where data is often stored in well-structured formats (e.g. tables) the field of big data generally includes the analysis of non-structured data (e.g. open texts) as well.

1.1 Brief History of (Big) Data

Since ancient times philosophers argue about wisdom and knowledge, often using these terms in a board, generic sense. The history of data in a more technical sense is tightly linked to the modern evolution of mathematics and IT technologies, which is shortly summarized in this section loosely following article [42].

Starting in year 1944 Fremont Rider, a Wesleyan University Librarian, in his paper "The Scholar and the Future of the Research Library: A Problem and its Solution" [43] estimates that American university libraries were doubling in size every sixteen years.

In year 1965 the US government plans the world's first data center to store approximately 740 million tax returns and 175 million sets of fingerprints [14]. Two years later, in 1967 the term "information explosion" is first used by Marron and de Maine in article "Automatic Data Compression" [38].

In year 1970 IBM researcher and mathematician Edgar F. Codd publishes "A Relational Model of Data for Large Shared Data Banks" [20]. In this he proposes a new concept, the *relational database*. During the same research Codd also creates the *Structured Query Language* (SQL) to easily modify, govern and query data in relational databases. In the following decade the concept of a data machine, a single mainframe computer system, becomes increasingly popular to handle data intense computations.

In the '80s a new concept emerges, the Teradata system, a "share nothing" parallel database system, where every machine has its own processor, storage and disk [24], [35]. In 1986 Teradata delivers the first parallel database system with a one terabyte storage capacity.

In 1996 Ralph Kimball publishes "The Data Warehouse Toolkit" pioneering the analytical database design named *star schema* based on its appear-

[2] 1 EB = 10^{18} B = 10^{9} GB

ance [34]. In the same year Morris and Truskowski show that digital storage became more and more cost-effective as time passes [39].

In 1997 Michael Lesk in article "How Much Information Is There In the World?" estimates that in the year 2000 we will be able to save in digital form everything we want to and later we may reach a world in which the average piece of information is never looked at by a human [36].

The term *big data* is first used in year 1999 in article "Visually exploring gigabyte data sets in real time" by authors Bryson, Kenwright, Cox, Ellsworth and Haimes in the title of section "Big Data for Scientific Visualization" [18]. The term big data is used again in the very same year in the title of panel "Automation or interaction: what's best for big data?" [33]. In the following year the term gains additional popularity, being used in the title of article [25].

In 2003 Google researchers Ghemawat, Gobioff and Leung present the Google File System [27], a scalable distributed file system for large data-intense applications aiming to provide fault tolerance on inexpensive commodity hardware. This concept revolutionized the field of big data, spearheading the era of distributed systems tackling large data sets. As a continuation of this research in year 2008 authors Dean and Ghemawat present the MapReduce programming model for processing large data sets [23].

During the following years big data becomes a widespread term, featuring in several newspaper articles, e.g. [21], [37] and a special issue of nature fully dedicated to this topic [10].

In 2009 Microsoft researchers Hey, Tansley and Tolle publish "The Fourth Paradigm" [30], where they describe how scientific breakthroughs will be powered by advanced computing capabilities.

This concludes the short timeline of data, data analysis and big data. During the decades SQL become the "lingua franca"[3] of data, sometimes being used for not fully relational database systems as well. In the past two decades the research area of big data and the amount of related technologies exploded as seen in the following chapters.

[3]bridge language, common language

1.2 Characterization of Data

Data come in different shapes and sizes. For centuries writing[4] was the golden standard of collecting and persistently storing data. Under favorable circumstances such writings are extremely durable and persistent. In the past century several millennia old documents resurfaced, including the famous Dead Sea Scrolls, approximately dating from the 3rd century BCE[5] [22].

In a simplest sense data is collected into *data sets*[6]. A *database* is an organized collection of structured and semi-structured data usually controlled by a *database management systems* (DBMS). A list of all students attending a class is an everyday example of data collected into a simple data set. A sheet of paper containing this data set in a structured format is a database, whereas the pen used to write the given list down is a commonplace form of a database management system enabling the capture and further modification of this simple database.

To simplify communication several characterization of data are used describing different aspects of data. While some of the following notions apply to non-digital data, this text mainly focuses on data stored in electronic form.

The first aspect concerning data is storage. Based on this aspect data can be divided into data at rest, in use and in transit. *Data at rest* are stored physically on persistent media either locally or on the cloud. This is the most durable form of data and is often subjected to additional archiving. Contrary to data at rest stands the term *data in transit*, sometimes referred to as *data in flight*. Data in transit is data transferred from a source to a destination using computer network [31]. Last, *data in use* complements the two options above, describing data stored on less durable, faster and non-persistent media, usually for computational purposes, e.g., memory units or caches.

Streaming refers to the action of transmitting continuous flow of data (known as *streams*) for a purpose of continuous processing. Individual data records often represent events, which may require timely response without a possibility to wait until all data are captured and transmitted.

Data can be divided based on their creation to machine-generated data, business application data and human-generated data. *Machine-generated*

[4] on papyrus, parchment, cloth or paper

[5] Before Common Era

[6] The term *dataset* is sometimes used interchangeably with data set, but it is not yet included in official dictionaries.

data are created by computer processes or sensors without human interaction. Usual examples of machine-generated data are application logs, network logs or sensor data. *Business application data* accumulate during the usage of particular applications supporting the business. An example of business application data is data from billing systems. *Human-generated data* are created by humans, such as word documents, images or social media interactions.

Data can be categorized as structured, semi-structured or unstructured, based on their internal structure. *Structured data* adhere to the principles of relational databases, i.e. they are organized into tables. In a relational database model related data are represented by tuples called *records* with a predefined set of *fields*, stored as rows in a table, where each field corresponds to a specific column. *Semi-structured data* are structured data, which do not have a tabular structure. Their structure is often described inside data using tags to annotate certain parts or features, the most well-known examples are XML and JSON. Last, the notion *unstructured data* refers to data without any apparent structure. The usual examples of unstructured data are open texts, audio, sensor data or Internet of Things data.

Certain data are subject to rules, agreements or law, which fact is collectively referred to as data confidentiality. The definition and characterization of confidential data differ between countries, enterprises and industries, therefore the following notions are provided in a generic sense. *Open data* are usually publicly accessible, often collected by *data stores*, repositories persistently storing and managing data. Other data might be *business confidential*. Such data are directly connected to business in one form or another and their loss can cause harm to the business in question, e.g., trade secrets or business decisions. *Personal information* or *personal data* (abbreviated to PI or PII) is any piece of information, which allows the identification of certain individuals. *Sensitive personal information* (SPI) or *sensitive personal data* is personal data concerning 'sensitive' subjects, such as race, origin, political opinion, sexual orientation, health-related information, genetic or biometric data, etc. Personal information, especially, sensitive personal information are subject to law and regulations, such as the famous EU legislation, named General Data Protection Regulation (GDPR)[4].

The term *metadata* refers to data about data. It provides additional information about data sets or individual data entries such as structure, size, source, purpose or legal restrictions. *Curated data* are annotated and organized after their collection or creation. These annotations are captured

in metadata, which are continuously managed during the whole life cycle of data, so that this information remains correct any given time.

1.3 Data Analysis and Analytical Systems

Data can be further categorized based on their purpose and use. Certain data facilitate the correct and efficient operation of business itself. These data and those systems handling them are called *operational*. *Analytical* data and systems are, on the other hand, used to provide insight into certain aspects of the business or predict possible and plausible future outcomes.

As an example, an e-shop is considered. During the placement of orders users of the given platform generate data, which are later used to fulfill given orders. Data generated by the order placement procedure is clearly operational in nature, being essential to fulfill orders *correctly* and *efficiently*. During the usage of the e-shop the users are prompted to give feedback on certain services provided by the platform itself. These data are later used to evaluate design choices and drive further development to achieve user satisfaction. These data are analytical in nature, providing *easy-to-understand evaluation* and *facilitate* business decisions.

From the example above it is easy to see, that analytical and operational data adhere to different requirement. In the case of operational data, efficient data access is often required to individual data entries for processing (read) and changing (update) alike. It is mandatory for operational systems to handle huge amounts of transactions concurrently. Analytical systems often provide aggregated results to highlight trends or verify assumptions under different circumstances. This requires an efficient way of reading data, especially in relation to aggregating and filtering large amounts of entries satisfying certain conditions. Results of these queries must be presented in an understandable form. The efficiency of creating (write) and maintaining (update) analytical data is a secondary concern, often handled by dedicated solutions periodically loading data into the analytical database in batches.

It is important to note, that any piece of information can have a dual role. In such cases the same information is usually captured and handled in two unrelated data sets, with very different structures and hosted inside distinct, dedicated environments. A prime example of such information are the orders in the e-shop from the previous example. On one hand, this information is captured into an operational database to facilitate correct functionality of the

e-commerce business. On the other hand, the same information is usually captured in an analytical system as well, providing customized recommendations for users.

In traditional relational databases operational data are usually collected and maintained in so-called *normal forms*. Normal forms ensure the correctness and integrity of data in an efficient manner[7].

Analytical databases became popular in the '90s of the previous century, with Ralph Kimball publishing his book 'The Data Warehouse Toolkit' [34] introducing the concept of *data warehouses* (DW) utilizing already established relational databases. In the same book he introduced *dimensional modeling*, a technique to create logical model for consistent analytical systems, supporting efficient aggregation and filtering.

Analytical data are often presented to end users via *reports* to enhance their understandability. Reports contain different representations of data, e.g., tables, charts, key performance indicators (KPI) in a human-readable format, presenting data consistently in a timely way. *Static* reports are a collection of findings presented to end-users as simple documents, while *dynamic* reports provide insight into data based on configurable input conditions. To achieve this functionality dedicated *business intelligence* (BI) tools are used to interact with the data warehouse. For this reason it is accustomary to abbreviate traditional analytical systems as DW/BI.

1.4 Big Data

In recent years huge amount of technologies, paradigms and techniques emerged to tackle the processing of ever growing volume of data. Same trends apply to data streaming, where the increase of volume of data transmitted had severe impact on analytical capabilities. These new techniques and technologies are often branded together under the umbrella of big data.

There were many attempt in the past to define big data. By its nature big data is an interdisciplinary notion related to IT and business alike. Under these circumstances the term "big data" became a "buzzword" used by industry experts as well as the general population. In what follows four definitions are presented and discussed to showcase the various ways authors tried to tackle the definition of the notion "big data".

[7] writing, updating and reading

In 2010 Apache Hadoop defined big data focusing on the size of data sets to define big data [19]. The definition itself is subjective, as it is never defined, what "acceptable scope" means.

Definition. Datasets which could not be captured, managed, and processed by general computers within an acceptable scope.

In June 2011 McKinsey published a report, referring to big data as "The next frontier for innovation, competition, and productivity" in the title of the report, focusing mainly on the outcome and potential of big data instead of its volume. In the same report the term big data is defined and explained similarly to the previous definition [28].

Definition. "Big data" refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze.

In article [26] big data is defined focusing mainly on the technological aspect of the field.

Definition. 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 the high-velocity capture, discovery, and/or analysis.

As a conclusion the definition of big data is vague and highly subjective. Despite this fact the term itself is useful to communicate intentions, expectations and considerations regarding data.

Another attempt to define big data based on their properties arose during the early years of the field. This definition was based on three particular property of big data, all starting with the letter 'V' leading to the naming *3V's*. In the following years additional important properties of big data were discussed and the *3V's*, now enriched by additional V's, evolved from a strict definition to a description of big data properties. In the following five properties of big data are presented collectively known as the *5V's*.

The first property of big data is its *volume*, i.e. size of data sets. Big data solutions and technologies should be able to process and analyze data on scale of petabytes, if needed.

The second property of big data is *variety*. Big data solutions often require to analyze structured and semi-structured sources together with open text documents, social media interactions or IoT sensor data.

Third, *velocity* means, that data is either generated fast in a streaming environment or data might loose its usefulness over time. Efficient processing and low response times are required in both cases.

Veracity of data tells how much certain data can be trusted. Not all data are inherently correct and not all sources of data are trustworthy. Even the machine generated data can become faulty under unforeseeable circumstances, but with the rise of social media the need to evaluate the correctness and validity of data is increasingly important.

Last, at the end of the data processing (or analysis) there must be *value* delivered to business. It is estimated that big data contains huge value, but with very low density. The main problem remains to identify this value and leverage it for the prospect of business.

The first industry widely utilizing big data solutions were IT companies focusing on the Internet since its early days. These companies had unique position in terms of skills, hardware capabilities and also motivation to handle huge volumes of data efficiently. Google widely utilized the MapReduce framework and the distributed filesystem they had developed, other companies, like Facebook, Twitter and Baidu following suit.

Other application of big data is in genom research and DNA sequencing, due to the huge volumes of data generated by sequencing. Healthcare and public health companies and high-frequency trading companies also need to analyze huge amounts of high-velocity data. Certain data warehouses grew significantly in the recent years, which lead to the migration of data warehouses to a big data toolkit to accommodate this growth.

Due to the increasing value of data, enterprises collect and retain data in a digital form for longer periods of time. A *data lake* is a central repository designed to securely store and process large amounts of data. Data lakes can store structured, semi-structured or unstructured data in their native format[8] or in a standardized form. A data lake is a central repository for a given enterprise, containing data for all of its teams and branches, serving as a *single source of truth* for the whole enterprise. Data within data lakes is usually curated and often standardized, to ensure validity and correctness.[16][29][32]

Creating a data lake poses a challenge. It is not uncommon to store raw data in the data lake without curation or governance. When such data build up the trustworthiness of the whole data lake becomes compromised, which situation is sometimes called *data swamp*. A different problem arises, when

[8] Also referred to as *raw data*.

a given enterprise starts building a data lake without any future outlook in mind. In this situation business has huge amounts of potentially important data, but they are not utilizing them, maintaining an expensive *data graveyard* in the process.[16][40]

When a new business case arises the data lake is searched for data supporting this new business case. This procedure is sometimes referred to as *exploration*. Data is usually explored by *data scientists*, sometimes relying on machine learning methods to achieve success.

It is important to note, that data warehouses and data lakes differ from each other both in structure and purpose. Data lakes are repositories of structured or unstructured data maintained for further use. On the contrary, data warehouses host structured data standardized and cleaned to support already existing business cases.

1.5 Wider Context

The second part of the 20th century was dominated by relational databases. With the increase of Web 2.0 - websites with user generated content - an old concept resurfaced, the non-relational database design, named *NoSQL*. While the name originally referred to non-SQL or non-relational databases, during the years its meaning changed to *Not only SQL* databases to emphasize that certain NoSQL databases support SQL like query languages as well.

There are several differences between relational and NoSQL databases. The most important difference is the fact, that NoSQL databases often relax some of the ACID properties[9] to achieve a flexible data model and scalability. NoSQL databases are mainly aimed to provide low-latency, while processing semi-structured data. There are many types of NoSQL databases, e.g., Key-Value databases, in memory Key-Value caches, Document Stores, graph databases.[17]

The simplest NoSQL databases operate on a data model known as dictionary, where data is stored in *key-value* pairs. Each key is unique in a dictionary and uniquely defines the set of attributes associated with it, its value. Examples of key-value databases are Amazon DynamoDB or Oracle NoSQL Database.[15][41]

A special case of key-value databases are *distributed in-memory caches*,

[9] Atomicity, Consistency, Isolation, Durability

providing fast response times utilizing memory for storing data. Examples of such in-memory caches are Memcached and Redis.[7][13]

Document stores allow developers to store data in a hierarchical store in forms of documents. These documents can range from XML or JSON documents to binary objects allowing the storage of documents in their native format. Documents can usually be tagged to allow more sophisticated filtering. An example of a document store is MongoDB[9], an example of a cloud based object storage is Amazon S3[1]. A special form of document store evolved from the specific use case of log collection and the consequent search of logs for error messages. Accommodating this type of workload the Elasticsearch engine became popular, currently being used for many other analytical use cases concerning huge amounts of open text or document searches.[3]

Certain data are represented in a form of graphs, where connections or interactions between nodes are an important feature to store and analyze. Such data are usually utilized by social networks or recommendation engines, amongst other use cases. *Graph databases*, such as neo4j and Apache Giraph, allow to store graphs efficiently.[2][11]

Early 21th century saw a big shift in terms of acquisition of infrastructure. For decades the golden standard of maintaining an infrastructure was to buy and physically own devices, needing constant maintenance. This methodology is currently challenged by a new paradigm, the so-called *cloud*, where users share computational power and other resources with each other. *Cloud providers* are entities owning and maintaining extensive amounts of physical hardware and leasing them to their clients, creating their own solutions using these resources. This new paradigm introduces a new challenge, the responsibility for the correct operation and maintenance of any given solution. Several service models are introduced by cloud providers to tackle this challenge, each of them providing *different split of responsibilities* between the cloud provider and the client.

Infrastructure as a service (IaaS) is the simplest form of cloud service models, where clients are provided with the lowest level of access and highest level of responsibility. In this case the client obtains a set of virtual resources from the provider in a form of virtual machines, storage, load balancers, etc. All of these virtual resources are fully maintained by the client.

Platform as a service (PaaS) is a model, where cloud providers create and maintain platforms from their resources to support certain type of workload. A good example of this service model is the large variety of ready-to-use database platforms. Using this service model the client leases and manages

the blank database instance of his choice (e.g. Oracle), while the provider manages the underlying infrastructure, which is shielded from the client altogether.

Software as a service (SaaS) is a model, where the cloud provider offers access to a pre-configured application running on cloud resources. Examples of this are various communication tools, such as email or video conference services. Another common example of the SaaS model are ready-to-use load balancer applications.

Function as a Service (FaaS), also called as *serverless computation*, is a split responsibility model, where the client is responsible only for the correctness of the code, while the whole execution environment is created and maintained by the cloud provider, usually in a temporary and self-contained manner. An example of the FaaS model is the Lambda function provided by AWS.

Cloud resources can be further categorized based on their operation. *Private cloud*, also called as *on-premises infrastructure*, is maintained by and dedicated to a single organization, located internally inside self-owned data centers or externally, managed by third party providers. The contrary of this operational model is *public cloud*, resources provided by public cloud providers, usually for a subscription fee. *Hybrid cloud* refers to solutions utilizing both public and private resources to achieve a common goal.

One of the biggest advantages of cloud environments is their *scalability*. To utilize this scalability applications running on cloud need to be designed with scalability in mind. Applications, which are designed to fit a cloud environment are sometimes referred to as *cloud native* applications and as a result they usually provide agility and speed backed by the cloud environment.

One of the sources of big data analysis is the so called *Internet of Things (IoT)*. The internet of things is a collection of interconnected sensors, processing resources and software that exchange data and is capable of autonomous reactions to the input data. Sensors in an IoT environment are a source of huge amount of high-velocity data.

References

[1] Amazon S3 Documentation. `https://aws.amazon.com/s3/`. [Online; accessed 06-Jun-2022].

[2] Apache Giraph. https://giraph.apache.org/. [Online; accessed 06-Jun-2022].

[3] Elasticsearch. https://www.elastic.co/. [Online; accessed 06-Jun-2022].

[4] General Data Protection Regulation. https://gdpr.eu/. [Online; accessed 06-Jun-2022].

[5] Global - 2021 Forecast Highlights. https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf. [Online; accessed 27-May-2022].

[6] International Data Corporation. https://www.idc.com. [Online; accessed 27-May-2022].

[7] Memcached. https://memcached.org/. [Online; accessed 06-Jun-2022].

[8] Merriam-Webster Dictionary. https://www.merriam-webster.com/dictionary. [Online; accessed 27-May-2022].

[9] Mongo DB. https://www.mongodb.com/. [Online; accessed 06-Jun-2022].

[10] Nature - Big Data. https://rdcu.be/cOwKI. Nature 455, 1 (2008). https://doi.org/10.1038/455001a.

[11] neo4j. https://neo4j.com/. [Online; accessed 06-Jun-2022].

[12] Oxford Learner's Dictionaries. https://www.oxfordlearnersdictionaries.com. [Online; accessed 27-May-2022].

[13] Redis. https://redis.io/. [Online; accessed 06-Jun-2022].

[14] R. S. Allen and P. Scott. Data Center Plan Called Privacy Invasion, 1966. Allen-Scott report; The Lewiston Daily Sun.

[15] AWS. https://aws.amazon.com/dynamodb/. https://aws.amazon.com/dynamodb/. [Online; accessed 06-Jun-2022].

[16] AWS. What is a data lake? `https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/`. [Online; accessed 03-Jun-2022].

[17] AWS. What is nosql? `https://aws.amazon.com/nosql/`. [Online; accessed 06-Jun-2022].

[18] Steve Bryson, David Kenwright, Michael Cox, Ellsworth David, and Robert Haimes. Visually exploring gigabyte data sets in real time, 1999. Communications of the ACM, Volume 42, Issue 8.

[19] Min Chen, Shiwen Mao, and Yunhao Liu. Big Data: A Survey. `https://doi.org/10.1007/s11036-013-0489-0`. Mobile Netw Appl 19, 171 – 209 (2014).

[20] E. F. Codd. A Relational Model of Data for Large Shared Data Banks, 1970. Communications of the ACM, Volume 13, Issue 6.

[21] Kenneth Cukier. Data, data everywhere. The Economist, Feb 27th 2010.

[22] Philip R. Davies. Dead Sea Scrolls. `https://www.britannica.com/topic/Dead-Sea-Scrolls`. Encyclopedia Britannica, [Online; accessed 30-May-2022].

[23] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters, 2008. Communications of the ACM, January 2008, Vol. 51, Issue 1.

[24] David DeWitt and Jim Gray. Parallel database systems: the future of high performance database systems, 1992. Communications of the ACM, June 1992, Vol. 35, Issue 6.

[25] Francis X. Diebold. "Big Data" Dynamic Factor Models for Macroeconomic Measurement and Forecasting, 2000. Eighth World Congress of the Econometric Society.

[26] John Gantz and David Rinsel. Extracting value from chaos, 2011. IDC iView, pages 1 - 12.

[27] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System, 2003. Symposium on Operating Systems Principles, ACM.

[28] McKinsey Global Institute. Big data: The next frontier for innovation, competition, and productivity. https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/big%20data%20the%20next%20frontier%20for%20innovation/mgi_big_data_full_report.pdf, 2011. [Online Report; accessed 02-Jun-2022].

[29] Google. What is a data lake? https://cloud.google.com/learn/what-is-a-data-lake. [Online; accessed 03-Jun-2022].

[30] Tony Hey, Stewart Tansley, and Kristin Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, October 2009.

[31] Julian Hyde. Data in Flight, 2010. Communications of the ACM, January 2010, Vol. 53, No. 1.

[32] IBM. Data lake solutions. https://www.ibm.com/analytics/data-lake. [Online; accessed 03-Jun-2022].

[33] David Kenwright, David Banks, Steve Bryson, Robert Haimes, Robert van Liere, and Sam Uselton. Automation or interaction: what's best for big data?, 1999. Proceedings Visualization '99 (Cat. No.99CB37067), Panel.

[34] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit - The Definitive Guide to Dimensional Modeling (Third Edition)*. John Wiley & Sons, Inc., 2013. ISBN: 978-1-118-53080-1.

[35] Teradata Labs. *Introduction to Teradata*. Teradata Corporation, 2016.

[36] Michael Lesk. How much information is there in the world? https://www.lesk.com/mlesk/ksg97/ksg.html. [Online; accessed 29-May-2022].

[37] Steve Lohr. The Age of Big Data. The New York Times, Feb 11th 2012.

[38] B. A. Marron and P. A. D. de Maine. Automatic Data Compression, 1967. Communications of the ACM, Volume 10, Issue 11.

[39] R. J. T. Morris and B. J. Truskowski. The evolution of storage systems, 1996. IBM Systems Journal.

[40] David Needle. Hadoop Summit: Wrangling Big Data Requires Novel Tools, Techniques. https://www.eweek.com/enterprise-apps/hadoop-summit-wrangling-big-data-requires-novel-tools-techniques. eWeek, 2015, [Online; accessed 03-Jun-2022].

[41] Oracle. Oracle nosql database. https://www.oracle.com/in/database/nosql/technologies/nosql/. [Online; accessed 06-Jun-2022].

[42] Gil Press. A Very Short History Of Big Data - Forbes. http://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/#446bb5de55da, 2013. [Online; accessed 27-May-2022].

[43] Fremont Rider. *The Scholar and the Future of the Research Library: A Problem and its Solution.* Hadham Press, 1944.