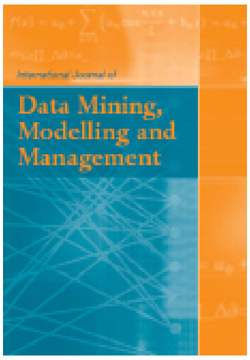


International Journal of Data Mining, Modelling and Management

ISSN online: 1759-1171 - ISSN print: 1759-1163
https://www.inderscience.com/ijdmmm

Developing a data pipeline solution for big data processing

Ivona Lipovac, Marina Bagić Babac

DOI: 10.1504/IJDMMM.2024.10058088

Article History:

Received:	27 August 2022
Last revised:	14 May 2023
Accepted:	16 May 2023
Published online:	22 January 2024

Developing a data pipeline solution for big data processing

Ivona Lipovac and Marina Bagić Babac*

Faculty of Electrical Engineering and Computing,
University of Zagreb,
Unska 3, HR-10000, Zagreb, Croatia
Email: ivona.lipovac@fer.hr
Email: marina.bagic@fer.hr
*Corresponding author

Abstract: This paper presents a comprehensive exploration of the concept of big data and its management while highlighting the challenges that arise in the process. The study showcases the development of a data pipeline, designed to facilitate big data collection, integration, and analysis while addressing state-of-the-art challenges, methods, tools, and technologies. Emphasis is placed on pipeline flexibility, with a view towards enabling ease of implementation of architecture changes, seamless integration of new sources, and straightforward implementation of additional transformations in existing pipelines as needed. The pipeline architecture is discussed in detail, with a focus on its design principles, components, and implementation details, as well as the mechanisms used to ensure its reliability, scalability, and performance. Results from a range of experiments demonstrate the pipeline's effectiveness in addressing the challenges of big data management and analysis, as well as its robustness and versatility in accommodating diverse data sources and processing requirements. This study provides insights into the critical role of data pipelines in enabling effective big data management and showcases the importance of flexibility in pipeline design to ensure adaptability to evolving data processing needs.

Keywords: big data; data pipeline; data processing; data analysis; cloud computing.

Reference to this paper should be made as follows: Lipovac, I. and Bagić Babac, M. (2024) 'Developing a data pipeline solution for big data processing', *Int. J. Data Mining, Modelling and Management*, Vol. 16, No. 1, pp.1–22.

Biographical notes: Ivona Lipovac received an MS degree in Computer Science from the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia. She works at Syntio Company as the Lead Data Engineer. Her professional and research interests include data engineering, and data science applications.

Marina Bagić Babac is an Associate Professor at the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia, where she obtained her DiplIng, MSc and PhD in Electrical Engineering. She also obtained an MSc in Journalism from the University of Zagreb, Faculty of Political Science. She is currently a collaborator on several international projects related to data science. She serves as a program committee member of a few international scientific conferences and journals, as well as a reviewer in numerous international journals. Her research interests include machine learning, natural language processing, and social network analysis.

1 Introduction

Big data refers to the large and complex datasets that are generated from various sources such as social media, sensors, and mobile devices (Manyika et al., 2011). The rapid growth of big data has created opportunities and challenges for businesses and scientific disciplines. Big data has been around for almost a decade (Stephenson, 2018); however, technical limitations have made big data storing, processing, and drawing insights from it, very difficult (Mayer-Schönberger and Cukier, 2013). At the start of the 21st century, Google made a big step toward big data handling by finding a way for computers to work together and form a cluster of machines with much more power than a normal computer. This was also a foundation for the development of Hadoop[1], one of the first distributed frameworks for data management. Modelled on Google, data-driven companies began developing their solutions to overcome big data challenges and laid the foundation for big data to become mainstream (Stephenson, 2018).

After the foundations were laid, the development of big data technologies continues more intensely, with new and improved technologies being introduced every day (McAfee and Brynjolfsson, 2012). The rapid development of technology encouraged the growth of the generated data. There are a few key reasons for such an increase in the generated and collected data. Firstly, the number of devices that generate digital data such as mobile phones, computers, sensors, etc. is constantly increasing. Secondly, modern technology enabled publishing content easy and available to everyone, which caused tremendous growth of human-generated content (Lazer et al., 2014). Finally, a constantly decreasing cost of digital storage enables the collection and storage of the generated data.

The collected data itself does not have much value until it is analysed and information is yielded from it (Segaran and Hammerbacher, 2009). The information gathered from the data brings a large spectrum of opportunities in numerous domains, such as making faster advances in different scientific disciplines, improving the success and profitability of companies, and reducing effort, time, and cost. The benefits and opportunities that big data brings, encouraged big data to be referred to as "the oil of the twenty-first century" (Segaran and Hammerbacher, 2009).

The purpose of this paper is to examine the challenges of big data processing and propose a data pipeline for efficiently handling big data, with a discussion on potential applications of our proposed solution. To efficiently handle big data, we propose a data pipeline that consists of several stages. The first stage is data acquisition, which involves gathering raw data from various sources (Chen et al., 2014). The second stage is data pre-processing, which involves cleaning and transforming the data to make it more usable (Katal et al., 2013). The third stage is data storage, which involves storing the processed data in a suitable format for efficient retrieval and processing. The fourth and final stage is data analysis, which involves applying analytical techniques to extract meaningful insights from the data (Chen et al., 2014).

Our paper presents a novel approach to building a data pipeline using Google Cloud services. We address several challenges that arise during the development of the pipeline, such as cost management and resource allocation, and propose solutions to overcome these challenges. Our approach involves scheduling of cloud functions using Cloud Scheduler, efficient use of Dataflow jobs, and careful management of data storage using Big Query. This paper contributes to the existing literature by providing practical solutions to common challenges in the development of data pipelines and demonstrates the effectiveness of using Google Cloud services for this purpose.

Our proposed solution has several potential applications in various fields that can be further developed and implemented to help businesses and organisations make more informed decisions based on the vast amount of data available to them.

2 Literature overview

With the growth of big data, interest in it grew, and thus the number of scientific papers on the subject related to big data. The information contained in big data has numerous benefits, so the question arises of how to get as much information as possible in the most efficient way. The answer to this question is not unique and there are numerous papers offering solutions (Kitchin, 2014).

Mobile network operators are a suitable example of a service that can benefit from big data. Simaković et al. (2021) focused on proposing a big data solution that collects and processes enormous amounts of data that mobile operators have for extracting valuable data which will help them improve their service. They proposed architecture with both batch and streaming support. The tools used for this solution are Apache Kafka[2], HDFS[3], Apache Spark[4], Apache Airflow[5], etc. (Chambers and Matei, 2018).

The focus of the paper by Suleykin and Panfilov (2019) was a workflow-based implementation of a data processing pipeline that is composed of open-source technologies. The proposed workflow, which is applicable in multiple domains, is constructed using Apache Spark for data processing, Apache Airflow as a scheduler, HDFS as a data storage layer, and PostgreSQL[6] as a serving layer.

Another domain that meets with huge amounts of data problems is a social science that gathers its data from multiple sources such as newspapers, social media, television, radio, etc. Singh et al. (2021) proposed a solution for social science researchers in a form of a portal for collecting, storing, and processing large-scale data sets. To ensure the reliability and scalability of the portal, Google Cloud Platform[7] tools are used. Cloud Storage is used for storing build artifacts, collected data, and computed aggregates, while PostgreSQL is used for storing metadata. Google's Dataproc[8] is used for running several jobs and Apache Airflow is used for scheduling and orchestration of the jobs. There were numerous challenges developers of this portal faced such as variance in data size, adjusting algorithms to fit their use case, remaining cost-effectiveness despite high requirements, etc.

In a paper by Liu and Iftikhar (2015), optimisations of the data pipelines were proposed in the form of partitioning and parallelisation. The pipeline dataflow is first partitioned into multiple execution trees according to the characteristics of the pipeline, then within an execution tree, parallelism and shared cache are used to optimise the partitioned dataflow. A problem with data processing pipelines is a cache that temporarily stores the data between two components of a pipeline. These caches can consume lots of memory. The solution presented in the paper which tackles this problem uses vertical partitioning on the dataflows, which enables components in the same partition to use the single shared cache to transfer data between them. Finally, the optimisation of the partitions is accomplished through pipelining and multi-threading (Liu and Iftikhar, 2015).

In addition, Aion (2016) investigated managing big data with Hadoop, Apache Storm, and Data Warehouse. It was found that these tools are a suitable solution for performing high-end real-time and near-real-time analytics. Apache Storm was also compared with Apache Spark to find a tool that works better with Hadoop. Results of the research showed that Apache Storm in combination with Hadoop and Data Warehouse can successfully overcome the challenges of real-time big data processing.

Another study (Sulova, 2021) analysed the technological aspects of digital transformation in logistics to propose a solution for big data management in the logistics industry. The proposed framework was built with Apache Hadoop open-source software. This paper also analyses existing solutions for big data processing in logistics and based on these solutions defines basic principles for constructing big data management architecture, which are adaptability and flexibility, scalability, automation, intelligence, security, and stability.

A use case of big data processing in farming is presented by Roukh et al. (2020). They developed a framework for monitoring crops in real-time which helps maximise productivity and profitability in farm and business operations. A lambda architecture, which consists of two branches, one for batch processing and the other for stream processing, is implemented in the framework. For the batch processing branch or batch layer, Apache Hadoop is used, and Apache Storm for the stream processing branch or speed layer. There are also experiments conducted and discussed to test the performance of the developed framework. Conducted experiments showed how the framework is doing big data processing in a considerable time and is of great importance for the farmers. However, its efficiency could be improved by adding new server nodes for the computing tasks in a parallel mode (Roukh et al., 2020).

Furthermore, Rehman et al. (2019) tackled the problem in industrial environments where enormous amounts of data are generated every day, but the information is not yielded from it due to systems design restrictions. As a solution, they introduced an industrial big data processing engine which is going to be an extension of existing infrastructure and will handle the processing of the generated data. The proposed solution consists of three layers: data ingestion layer, data storage layer, and data processing layer. Apache Sqoop was used in the data ingestion layer for gathering data from loggers and servers and making it available on a distributed file system. HDFS was used as a base for the data storage layer as it provides fast and fault-tolerant data access to the last layer. Finally, the Python library Dask was used for performing computations on the data. The proposed engine was tested on different sizes of data and proved to be an effective extension of the existing system.

One recent study by Zhang et al. (2020) proposed a distributed data processing framework that utilises Apache Spark for data processing and Google Cloud Storage for data storage. Their study showed improved efficiency and scalability when compared to traditional Hadoop-based solutions. Another study by Bezerra et al. (2019) proposed a cloud-based data processing platform that uses container technology for managing data processing tasks. Their study demonstrated the benefits of using container technology, such as faster deployment and increased resource utilisation. In a similar vein, Zaman et al. (2021) proposed a cloud-based big data processing platform that uses container technology and Apache Spark for data processing. Their study focused on optimising resource allocation and showed improved efficiency and scalability when compared to traditional solutions. Furthermore, a study by Liu et al. (2014) proposed a data processing framework that uses Google Cloud Dataflow for data processing and storage. Their study demonstrated improved efficiency and cost-effectiveness when compared to traditional solutions.

There have been several studies that have evaluated the effectiveness of different cloud computing platforms and tools in the development of data pipelines. For example, Rehman et al. (2019) compared the performance of Google Cloud and Amazon Web Services in the development of a big data pipeline. They found that Google Cloud outperformed Amazon Web Services in terms of speed, cost-effectiveness, and ease of use. Similarly, Zhang et al. (2023) evaluated the effectiveness of Apache Airflow and Apache Nifi in the development of a data pipeline for processing web log data. They found that Apache Airflow was more effective in terms of scalability and cost-effectiveness.

In terms of the challenges faced during the development of our data pipeline, some of the issues, such as cost management, have been highlighted in previous studies. For example, Zhelev and Rozeva (2017) discussed the importance of cost management in the development of data pipelines and proposed a framework for optimising costs. Similarly, Jiang et al. (2020) highlighted the importance of resource allocation and scheduling in the development of data pipelines and proposed a scheduling algorithm to optimise resource utilisation.

Our proposed solution builds on these existing studies and extends them by utilising a data pipeline approach that leverages several Google Cloud services to address common challenges in the development of big data processing pipelines. In addition, our study highlights the importance of cost management and efficient resource allocation in the development of data pipelines. While there are several cloud computing platforms and tools available for developing data pipelines, careful consideration of costs and resource utilisation is essential to ensure the success of the pipeline.

3 Challenges of big data processing

The term big data is often described in the literature as datasets and collections of datasets whose complexity, size, and rate of expansion require such processing capabilities that go beyond the processing capabilities of traditional data management systems and software tools (Hussien, 2020). A similar definition can be found in Gartner’s IT 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”. (Gartner, 2022) This definition was first introduced by analyst Doug Laney and describes big data with 3V characteristics: Volume, Velocity, and Variety. With the continuous expansion of big data, the number of V characteristics describing big data also increased, and the 4 Vs of big data, the 5 Vs, and the 8 Vs definitions were introduced. The most recent proposed definition uses 56 Vs for describing big data (Hussien, 2020). Other characteristics such as veracity, value, variability, and complexity are needed to fully understand and utilise big data. Veracity refers to the accuracy and quality of the data, while value refers to the potential benefits and insights that can be derived from it. Variability refers to the inconsistency and diversity of the data, and complexity refers to the difficulty in managing and analysing it. All these characteristics play a role in determining the usefulness and relevance of big data in various contexts.

Volume refers to the immense amounts of data being collected every day which needs to be stored and processed. The velocity refers firstly to the speed of data growth and secondly to the speed of data transfer (Alguliyev et al., 2017). Variety mainly refers to the different forms of the generated data, which includes structured, semi-structured, and unstructured data. Each of these forms of data has multiple subforms. For example, unstructured data includes images, videos, textual data saved in multiple formats, etc. Unstructured data is the most common form since it contributes 90% to the total share of generated data[9].

There are challenges to big data, ranging from collection, management, and processing, to making sense of big data to drive business results. According to Sharma et al. (2017), big data challenges include capture, storage, analysis, data curation, search, sharing, transfer, visualisation, querying, updating, and information privacy.

3.1 Data collection

One of the major challenges of big data processing is the sheer size of the data sets. These data sets can contain millions or even billions of data points, which can be difficult to process using traditional methods (Chen et al., 2014). Additionally, big data sets are often unstructured, meaning that they do not conform to a standard format, which can further complicate processing (Katal et al., 2013).

The need to manage the ever-increasing amount of data has spurred the rapid development of computer resources. However, data volume is scaling faster than compute resources (Shakhovska, 2017). Since the computation of big data requires lots of resources and effort because of its size, it is necessary to decide which data is required for the purpose and which data needs to be discarded. This process is called data profiling and refers to examining data for its characteristics and structure and evaluating how well it fits a business purpose. Data profiling is an essential part of the data collection phase.

Big data is generated by numerous different devices in various formats and for that reason, the big data collection process is demanding. One of the challenges arising in this phase of big data processing is the heterogeneous data problem. Heterogeneous data implies that collected data comes in various sizes and formats. Different sources might use various data types and structures for representing similar information and this leads to collecting the same information represented in several different ways (Domitran and Bagić Babac, 2021).

Another challenge arising is the incompleteness of the collected data meaning that data is missing some fields or records. This type of data needs to be specifically handled by substituting, regenerating, or skipping from the pipeline. If not handled properly, missing data or inconsistent data can negatively affect the analysis and the results.

Traditional computer systems are not sufficient to perform analysis on this semi-structured data anymore (Thabet and Soomro, 2015). Therefore, heterogeneous and incomplete data must be structured data before analysis since computer systems and machine learning algorithms expect homogeneous data (Shakhovska, 2017).

A part of the collected data includes personal data such as locations, identities, health details, etc. A major concern rising with personal data is the inappropriate use of the same. There are questions rising around which entities have the right to collect this kind of data as well as users not being informed enough about how much of their personal data are they sharing. The privacy problem is sought to be overcome by the introduction of new laws which govern the privacy of users. However, not only personal data is at risk.

Many companies have enormous data repositories with user and company data and these repositories, if not secured correctly, can become a target for various criminal groups (Singh et al., 2021).

There is a security risk when collecting data from various sources as many of them might provide unauthentic data. This can be prevented by reviewing the sources and permissions (Sharma, 2018).

Previously mentioned privacy and security risks showed how smart data governance must be present in every phase of the data flow, from the data collection, and the transfer of the data, to storing the data. Finally, data governance guarantees data consistency and accuracy, which is necessary for making business decisions based on the data (Abualkishik, 2019).

3.2 Data integration

Data collected from various sources is transferred to storage. It is crucial to find the appropriate storage location since the data transfer to storage and fetching data for analysis demands lots of resources and energy. Another important aspect of data storage is scalability, respectively the ability of storage to adapt to the ever-changing nature of big data.

Another challenge occurring in the integration phase of big data processing is the challenge regarding metadata. The collected data in some situations is collected without belonging metadata, which makes data almost impossible to further analyse since metadata describes the data recorded and how it is recorded and measured. The data needs to be collected and transferred with appropriate metadata to analyse it qualitatively and yield insights from it. The thing that is not so intuitive, but is of great importance, is that the metadata needs to be generated during data processing as well. Recording the origin of the data and the movement of the data in the processing pipeline will help to determine the next processing steps since every step is dependent on the previous step (Zhelev and Rozeva, 2017).

The efficiency of the analysis depends on the quality database organisation. Designing the storage which fits the purpose of the data is of great importance since similar data can be differently organised for different domains (Thabet and Soomro, 2015). The process of creating an effective schema and mapping data is still being studied and improved to give the best results (Abualkishik, 2019).

In addition to the database organisation, one of the most common challenges of the big data integration phase is preparing the data for storage in the data warehouse. Aggregating, merging, and corroborating data collected from various sources can be time and cost-demanding (Blagaić and Bagić Babac, 2022). Predefined protocols and transformations are used for handling numerous data formats and different qualities of the collected data.

3.3 Data analysis

The analysis phase is the most complex phase regarding challenges that occur. The analysis process of big data consists of multiple subphases including extraction of information and cleaning, data aggregation and representation, query processing and data modeling and analysis, and interpretation and presentation (Thabet and Soomro, 2015).

The most obvious challenge of the analysis phase is the velocity of the data. The larger the data set to be processed, the longer the analysis will take. So, designing a system that will effectively deal with the enormous size of the data demands unique hardware and software utilities. If a system can handle the size of the data, then it will likely be able to process this data much faster as well (Shakhovska, 2017).

Streaming data is a particular challenge. This kind of data is short-lived and needs to be analysed in near real-time to yield insights from it (Sulova, 2021). Designing a system with a capacity for handling and analysing continuously generated data requires resources and effort (Cvitanović and Bagić Babac, 2022).

The extensiveness of the analysis depends not only on the amount of data but on the complexity of individual data. A single data can consist of thousands of variables and it is impossible to create a model which will be able to process all of the variables. For that reason, the analysis of big data must include the process of extracting the most important variables. Collected data might contain fake information which negatively affects both the analysis and the results of the analysis. For that reason, data-cleaning techniques need to be performed prior to analysis. However, that requires developing quality-control models which will validate the collected data. Another important aspect of data analysis is correlations and links between the data. It is necessary to create correct correlations among data to correctly interpret the data and for analysis to be successful (Puh and Bagić Babac, 2022).

Since new data is generated every second, it is important to decide when the data is no longer applicable for analysis. Performing analysis on data that is stored for too long in different storage might make the analysis not reliable. Speaking of reliability, it is also important not to rely completely on the results of the analysis, but to consult domain experts, apply common sense and react to outside events (Thabet and Soomro, 2015).

3.4 Everyday application

Big data processing is a very complex task that not only requires computation resources but qualified human resources as well. Many companies struggle to find qualified data engineers, analysts, and scientists who understand big data and big data processing and can successfully manage it (Lipovac and Bagić Babac, 2021).

There are various errors that can emerge in big data such as errors in generating data or errors in computer systems or measurements. These errors can propagate to the results of the analysis as well. For this reason, it is necessary for the results of the analysis to be verified by qualified experts (Bagić Babac and Podobnik, 2016).

Data analysts and decision-makers must be familiar with the assumptions that were used in each phase of the data pipeline to guarantee proper and valid decisions.

Another challenge that occurs with the application of the final knowledge yielded from big data is the complexity of the results. The results of the analysis can be usually interpreted by domain experts, but other users might not understand the results. One possible solution for this problem is to use visualisation tools as well as software that enable users to change variables and monitor how these changes reflect in the result (Poch Alonso and Bagić Babac, 2022).

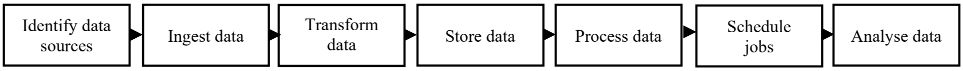
4 Data pipeline for big data processing

A data pipeline[10] is a set of tools and processes used for automating the movement and transformation of data from a source to a target repository. There are numerous types of data pipelines depending on the purpose of the big data processing, but generally, a data pipeline consists of four main steps: data acquisition, integration, data analysis, and real-world application. A data pipeline for big data processing typically involves multiple steps, including data ingestion, data processing, data storage, and data analysis.

Here are the general (sub)steps involved in building a data pipeline for big data processing based on the Google Cloud Platform (Figure 1):

- *Identify data sources*: The first step is to identify the data sources that will be used in the pipeline. These can include structured data sources like databases and unstructured data sources like log files.
- *Ingest data*: Next, the data needs to be ingested into the pipeline. This can be done using tools like Apache Kafka or Google Cloud Pub/Sub.
- *Transform data*: Once the data is ingested, it needs to be transformed into a format that is suitable for analysis. This can involve cleaning, filtering, and aggregating the data.
- *Store data*: The transformed data needs to be stored in a data warehouse or data lake. Google Cloud Platform provides several options for storing big data, including BigQuery and Cloud Storage.
- *Process data*: The data then needs to be processed. This can be done using tools like Apache Spark or Google Cloud Dataflow.
- *Schedule jobs*: Data processing jobs need to be scheduled to run on a regular basis. This can be done using tools like Apache Airflow or Google Cloud Composer.
- *Analyse data*: Finally, the processed data can be analysed using tools like Google Data Studio or Tableau.

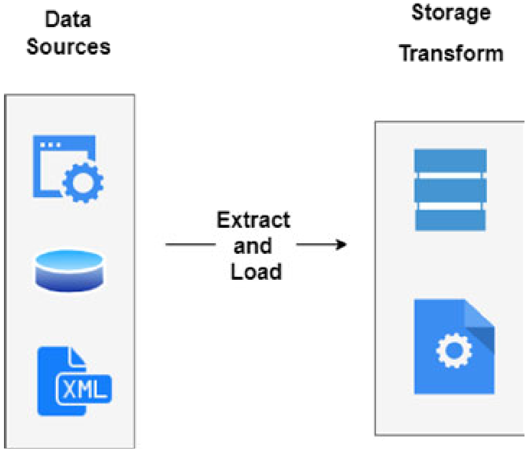
Figure 1 Big data processing pipeline methodology



4.1 ELT pipeline

The *extract, load, and transform* (ELT) pipeline includes extracting data, loading it into the target destination, and performing transformations within the target system (Figure 2). By performing the loading phase before the transformation phase, the ELT pipeline offers the ability to run transformations on the processing engines within the target system. ELT separates these two phases and offers additional flexibility in performing future changes to the data warehouse. The separation of the transformation and loading phase also simplifies the management of a pipeline and reduces risk by reducing dependencies. ELT pipeline is appropriate for transforming large datasets containing both structured and unstructured data.

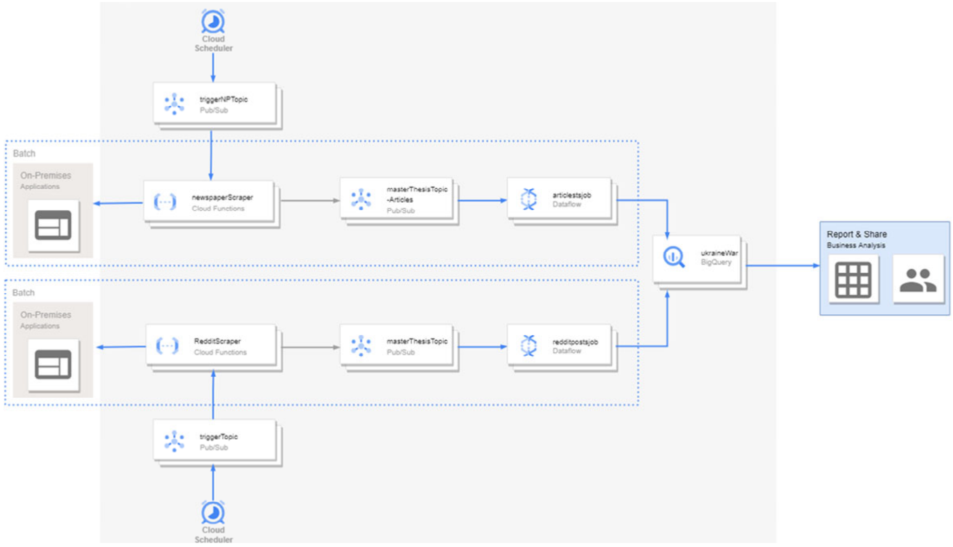
Figure 2 An ELT process (see online version for colours)



4.2 Data pipeline development

To demonstrate how some of the above-mentioned tools, technologies, and methods work, as well as to show the challenges occurring in the processes of gathering data, data transformation, and data storage, an ETL pipeline was developed.

Figure 3 The architecture of a data pipeline (see online version for colours)



4.2.1 Data pipeline architecture

The architecture of an ETL pipeline is conceived as follows; in the first stage, the data is extracted from different sources on a schedule on an hourly or daily basis depending on the type of the source. Once the data is extracted, it is published as messages on a predefined topic. The next two stages include a job that consumes data as messages from a topic, transforms this data to fit the defined schema of the storage, and loads the data into the storage. Finally, in the last stage, an analysis of the stored data is performed, and the results are reported. The architecture of a data pipeline is shown in Figure 3.

4.2.2 Tools, technologies, and methods used

Google Cloud Platform is chosen as a cloud provider as it offers numerous useful solutions for easily developing an efficient data pipeline. It also serves as a host for the different functions and jobs required for the purposes of developing a functional data pipeline.

To create resources on a Google Cloud Platform, a project is created. Additionally, for the authentication for the Google Cloud Platform services, a service account is created and used for all the services and tools required for the data pipeline.

The first stage of a developed ETL pipeline is data extraction from two different sources. Here, information regarding the war in Ukraine was gathered. The first source is a social network, Reddit, and the most popular posts regarding the war in Ukraine are collected on an hourly basis using the Python *praw*[11] library. The second source is NBC news articles from the 'Russia-Ukraine-news' section, which is collected using the Python *newspaper*[12] library. The articles were collected daily.

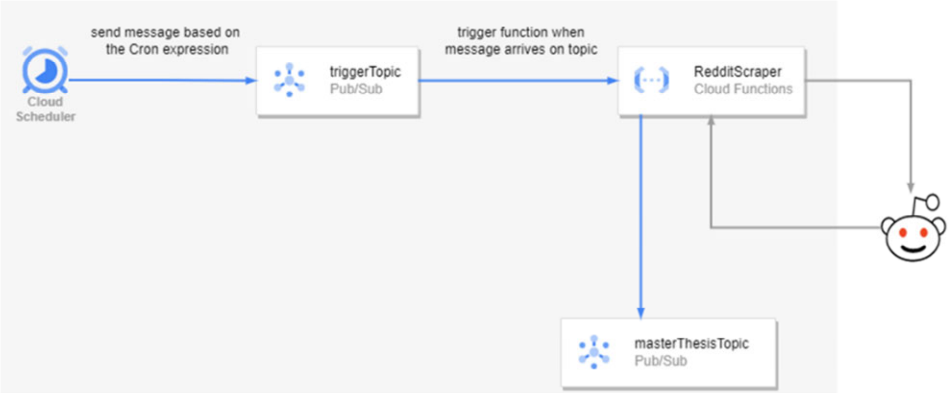
The Reddit scraper and the NBC news articles scraper were written in Python and deployed as Cloud Functions on a Google Cloud Platform. Cloud Functions are Google Cloud *function-as-a-service* (FaaS) product that gives developers the ability to run their code in the cloud, while Google manages the operational infrastructure. To schedule these Cloud functions to trigger on a daily or hourly basis, Cloud Scheduler is used. Cloud Scheduler is a fully managed *cron*-job scheduler that allows developers to run their code on a recurring schedule with an execution warranty as well as a warranty of retry in case of an error.

Another service needs to be mentioned before describing the Cloud functions scheduling, Pub/Sub. Pub/Sub is a producer-consumer service that enables the producer (also called a *publisher*) to send an event to a Pub/Sub service, and a consumer (also called a *subscriber*) to consume the delivered events.

There are two jobs defined in a Cloud Scheduler, *newspaperScraperTrigger* and *redditScraperTrigger*. These jobs communicate with Cloud functions, *newspaperScraper,* and *RedditScraper* through a Pub/Sub. In this case, Cloud Scheduler jobs have the role of publishers and Cloud Functions are consumers. Each Cloud Scheduler job publishes messages on a schedule on a predefined Pub/Sub topic. There is a topic defined for each job, *triggerTopic* for the Reddit Cloud Function and *triggerNPTopic* for the newspaper articles Cloud Function. Cloud Functions, which are consumers, wait for these events, each function waits for an event on a particular topic. When the event arrives, the Cloud Function is executed.

Cloud Functions scrape the data from the defined sources and form the data as JSON objects with predefined attributes. Once the data is scraped, it is published on the different Pub/Sub topics depending on the source, *masterPaperTopic* for Reddit posts and *masterPaperTopicArticles* for the NBC articles. In this last part of the data collection phase, the Cloud Function has a role of a Pub/Sub publisher (Figure 4).

Figure 4 The architecture of the data ingestion phase (see online version for colours)

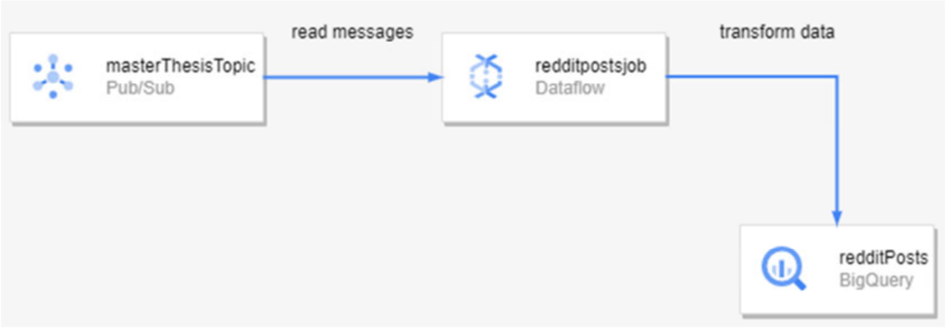

send message based on the Cron expression
trigger function when message arrives on topic
Cloud Scheduler
triggerTopic
Pub/Sub
RedditScraper
Cloud Functions
masterThesisTopic
Pub/Sub

Once the data is published on the topics, a consumer for the published data is defined. This consumer will read the data, transform it to fit the model of the storage and load the data into the storage. A tool that was the best fit for defining a consumer, is a Google Cloud Dataflow and Apache Beam. Google Cloud Big Query was chosen for data storage. The architecture of this phase of a data pipeline is displayed in Figure 5.

Two consumers have been developed, each for one source and one topic. A consumer is written in the form of a Beam pipeline using Python programming language and client libraries for different Cloud services. Finally, the consumer is deployed as a job in Cloud Dataflow. This is a fundamental part of the proposed ETL pipeline.

Dataflow is a Google Cloud-managed service for running batch and streaming data processing pipelines created with Apache Beam SDK. Dataflow is a fully managed service with support for autoscaling for resource utilisation and exactly one processing, which makes it a great tool for running cost-effective data pipelines. Big Query is a Google Cloud data warehouse that enables storing and managing data while the infrastructure is fully managed by Google Cloud.

Figure 5 The architecture of the transformation and data-storing phase (see online version for colours)



To run a Beam pipeline as a Dataflow job, pipeline options must be specified. Additional resources such as a folder for temporary files must be created prior to running a job. Options are described in Table 1.

Table 1 Pipeline options list

Option	*Description*
runner	Option determining the PipelineRunner at runtime
project	The project ID for the Google Cloud project
job_name	The name of the dataflow job being executed
temp_location	Cloud storage path for temporary files
region	Specifies a regional endpoint for deploying the dataflow job
streaming	Option for enabling streaming engine

Defining pipeline options in a code is shown below.

Code 1 Beam pipeline options

```
beamOptions = PipelineOptions(
 beam_args,
 runner='DataflowRunner',
 project='masterpaperproject-346710',
 job_name='articlesjob',
 temp_location='gs://temp_bucket_mt/temp',
 region='europe-west1',
 streaming=True,
 save_main_session = True)
```

Depending on a workflow of a pipeline, additional arguments are defined, and their values are passed when running a pipeline. In this case, the input Pub/Sub topic from which the data is read and the destination Big Query table where data will be stored, need to be defined. The code 2 example below shows how to define arguments expected at runtime.

Code 2 Defining arguments to be passed on a runtime

```
parser = argparse.ArgumentParser()
parser.add_argument('--input_topic', required=True, help='The
pubsub topic.')
parser.add_argument('--table', required=True, help='The BigQuery
table.')
```

There are three main phases defined in a Beam pipeline: reading messages from Pub/Sub topics, transforming data to fit the schema of the storage, and loading the data into the Big Query storage.

Code 3 Steps of a Beam pipeline

```
from apache_beam.options.pipeline_options import PipelineOptions
from apache_beam import DoFn, GroupByKey, io, ParDo, Pipeline,
PTransform, Map, Filter
```

```
# Create the Pipeline with the specified options.
with beam.Pipeline(options=beamOptions) as pipeline:
  (pipeline
  |"Read from Pub/Sub" >>
io.ReadFromPubSub(topic=args.input_topic)
  | "To Dict and Append Timestamp" >> Map(addTimestamp)
  | "Convert Created to Timestamp" >> Map(convertDate)
  | "Filter by Date" >> Filter(isTodayPost)
  | "Write To BigQuery" >>
io.gcp.bigquery.WriteToBigQuery(table=args.table))
```

The transformation consists of three steps: adding a timestamp with the date of storing the data in the Big Query, converting the time of generating the data into the preferred format, and filtering messages by date so only data generated on the current date is stored. Below is the code for the transformation functions.

Code 4 Transformation functions

```
def addTimestamp(message):
 data = json.loads(message)
 timestamp = datetime.now()
 data.update({"collected": timestamp})
 return data
def convertDate(data):
 timestamp = datetime.fromtimestamp(int(data['created']))
 data['created'] = timestamp.strftime('%Y-%m-%d %H:%M:%S')
 return data
def isTodayPost(post):
 if(post['created'] is not None):
 createdDate = datetime.strptime(post['created'], '%Y-%m-%d
%H:%M:%S').date()
 todayDate = datetime.today().date()
 if (createdDate == todayDate):
 return True
 else:
 return True
```

Before running the pipeline, destination tables need to be created. Two tables are created, *articles* and *redditPosts*. Tables are first created using Big Query UI. The ID of the created table is provided in the script, where the schema for the table is defined and created. This is achieved with two simple Python scripts using the Python client library for Big Query.

The integer score of a Reddit article represents the number of upvotes minus the number of downvotes it has received (the 'score' parameter in the following code). When a user upvotes a post, its score increases by one, and when a user downvotes a post, its score decreases by one. The score is used as a way of ranking the posts on the site, with higher-scoring posts being more visible to users (Bagić Babac, 2022).

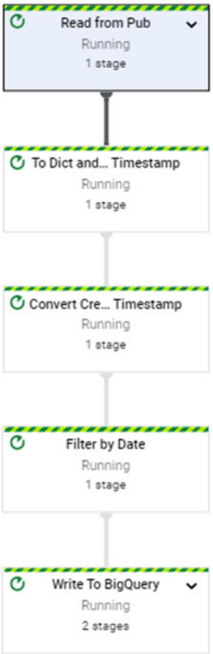
Code 5 Defining and creating table schema

```
from google.cloud import bigquery
client = bigquery.Client()
tableId = “masterpaperproject-346710.ukraineWar.redditPosts”
schema = [
 bigquery.SchemaField(“title”, “STRING”, mode=“REQUIRED”),
 bigquery.SchemaField(“score”, “STRING”, mode=“REQUIRED”),
 bigquery.SchemaField(“id”, “STRING”, mode=“REQUIRED”),
 bigquery.SchemaField(“url”, “STRING”, mode=“NULLABLE”),
 bigquery.SchemaField(“comms_num”, “BIGNUMERIC”,
mode=“NULLABLE”),
 bigquery.SchemaField(“created”, “STRING”, mode=“REQUIRED”),
 bigquery.SchemaField(“body”, “STRING”, mode=“NULLABLE”),
 bigquery.SchemaField(“collected”, “STRING”, mode=“NULLABLE”),]
table = bigquery.Table(tableId, schema=schema)
table = client.create_table(table) # Make an API request.
```

Finally, when the Beam pipeline and options for running a pipeline inside of a Dataflow job are defined, a script can be executed with the command:

```
python pipeline.py --input_topic=projects/masterpaperproject-
346710/topics/masterPaperTopic --table=masterpaperproject-
346710:ukraineWar.redditPosts
```

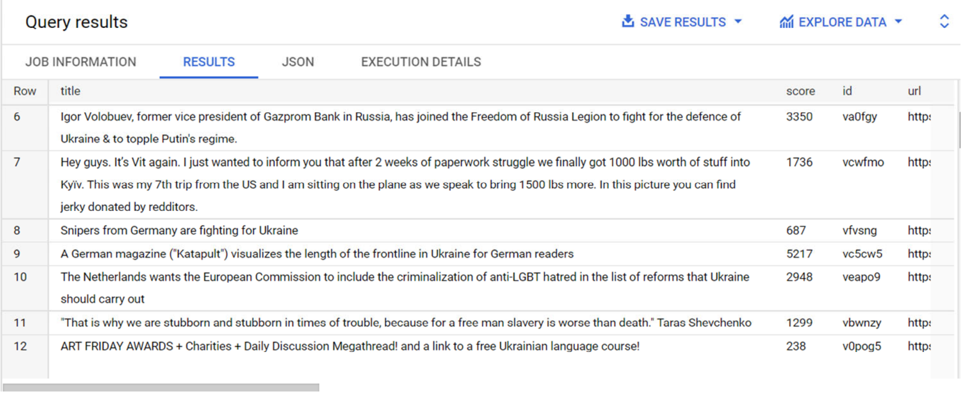
Figure 6 Dataflow job represented as a graph (see online version for colours)



By starting a dataflow job, a data pipeline is ready for transferring data from sources to destinations. Figure 6 shows the Dataflow job represented as a graph divided into stages.

Collected Reddit posts are visible in Big Query tables shown in Figure 7.

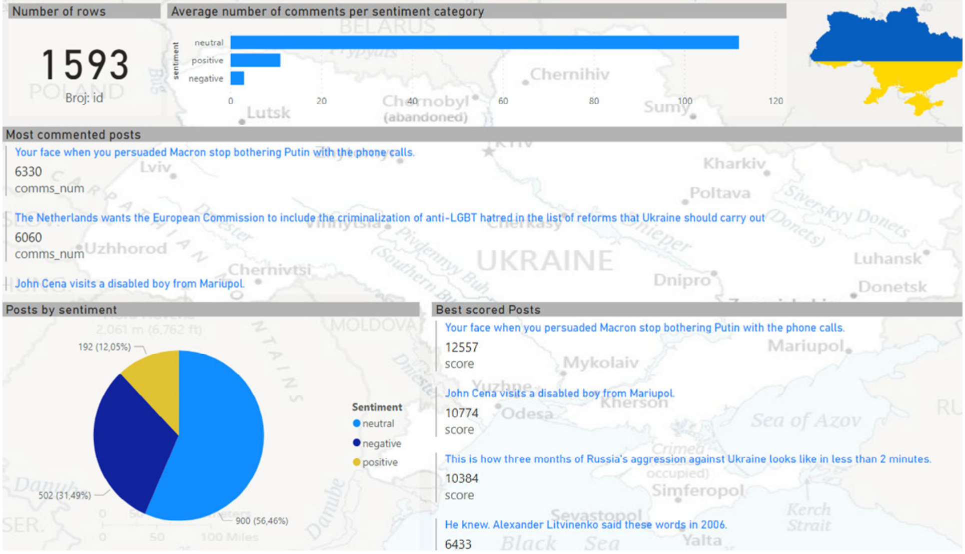
Figure 7 Collected NBC articles saved in a Big Query table (see online version for colours)



Row	title	score	id	url
6	Igor Volobuev, former vice president of Gazprom Bank in Russia, has joined the Freedom of Russia Legion to fight for the defence of Ukraine & to topple Putin's regime.	3350	va0fgy	http:
7	Hey guys. It's Vit again. I just wanted to inform you that after 2 weeks of paperwork struggle we finally got 1000 lbs worth of stuff into Kyïv. This was my 7th trip from the US and I am sitting on the plane as we speak to bring 1500 lbs more. In this picture you can find jerky donated by redditors.	1736	vcwfmo	http:
8	Snipers from Germany are fighting for Ukraine	687	vfvsng	http:
9	A German magazine ("Katapult") visualizes the length of the frontline in Ukraine for German readers	5217	vc5cw5	http:
10	The Netherlands wants the European Commission to include the criminalization of anti-LGBT hatred in the list of reforms that Ukraine should carry out	2948	veapo9	http:
11	"That is why we are stubborn and stubborn in times of trouble, because for a free man slavery is worse than death." Taras Shevchenko	1299	vbwnzy	http:
12	ART FRIDAY AWARDS + Charities + Daily Discussion Megathread! and a link to a free Ukrainian language course!	238	v0pog5	http:

To demonstrate how the gathered data can be used for gaining important insights, a Power BI report is created. Power BI[13] is a collection of software tools that enables connecting to multiple data sources and storage and visualising the data to extract information from it.

Figure 8 Power BI Reddit posts report (see online version for colours)



Most commented posts

Your face when you persuaded Macron stop bothering Putin with the phone calls.
6330
comms_num

The Netherlands wants the European Commission to include the criminalization of anti-LGBT hatred in the list of reforms that Ukraine should carry out
6060
comms_num

John Cena visits a disabled boy from Mariupol.

Best scored Posts

Your face when you persuaded Macron stop bothering Putin with the phone calls.
12557
score

John Cena visits a disabled boy from Mariupol.
10774
score

This is how three months of Russia's aggression against Ukraine looks like in less than 2 minutes.
10384
score

He knew. Alexander Litvinenko said these words in 2006.
6433

Then, a report based on collected posts from Reddit was made. Power BI is a powerful tool that offers numerous features for data manipulation. The report in Figure 8 includes the number of records when making a report, posts ordered by the highest score, and posts ordered by the number of comments. Also, simple sentiment analysis is performed on the posts. This is another useful feature of Power BI, which enables users to write Python or R scripts and execute them on the imported data.

The results of the sentiment analysis (Marijić and Bagić Babac, 2023) show that neutral posts are the most frequent, followed by negative posts and the least frequent positive posts. Also, an average number of comments per sentiment category is also displayed and it shows that although negative posts are more frequent than positive posts, positive posts are more commented on, unlike negative posts.

4.2.3 Challenges discussion

There are several challenges that occurred during the development of the data pipeline which should be discussed. A data pipeline is built using Google Cloud services which are being charged. For the purposes of this study, Google Cloud free trial was used which gives users $300 and 90 days to use services for free. After that, services are charged. So, the main challenge was to not exceed the mentioned limit. For that reason, it was necessary to choose which services were going to be used. For example, scheduling of cloud functions could be done using Cloud Composer, which is based on Apache Airflow. However, Cloud Scheduler was used instead as it is more cost-effective, and it fits the purpose. Another thing that had to be considered, is running Dataflow jobs. These jobs could not run all the time because that would incur increased costs. Therefore, Dataflow jobs were shut down from time to time. The last thing regarding costs is Big Query. It was important not to put too many records on the tables. For that reason, scrapers were scheduled on an hourly and daily basis and were limited to scraping five posts and five articles.

Another limitation is related to Power BI. Namely, Power BI offers a feature ‘scheduled refresh’, which enables users to schedule the refreshing of the dataset and of the report as well. This feature is not possible in Power BI Desktop based on the personal account, which was used for the purposes of this study. However, there is an option of performing a manual refresh of the dataset and the report, and this option was used.

When it comes to challenges regarding methods, the challenge that needs to be mentioned is data duplication, which is a very common challenge. There are multiple reasons behind duplicate data. For example, duplicate data can be generated by the source. Another reason could be the publisher, in this case, Cloud Function, which scrapes data and publishes it on the Pub/Sub topic. For example, if no new articles are published on the website, there is a risk that several old articles will be scraped and published on the topic. Also, if message publishing was interrupted due to network issues or for other reasons, Pub/Sub will retry publishing the message which ends up with duplicate data (Khan, 2021).

There are methods for handling duplicate data. One way is to integrate Apache Beam PTransforms, which can deduplicate data over a time duration. Deduplication is performed based on a key-value pair from the incoming message. However, this was not a suitable option for implementing a data pipeline, since there was a longer time interval between batches of messages. Therefore, another approach had to be considered. Finally, deduplication was performed partly in Dataflow and partly in storage, Big Query.

In Dataflow, there was a deduplication function implemented as Apache Beam Filter, which filtered messages by the date, so only messages from today’s date were sent to the storage. This only partly deduplicated messages. Therefore, additional deduplication is performed in the Big Query as a query that removes duplicates from the latest part of the table. This query is scheduled to run accordingly to the Cloud Functions which scrape the data from the websites. So, as soon as new data is loaded into the Big Query table, a scheduled query is executed on the new data and duplicate records are removed.

To conclude, the developed ETL pipeline effectively transports data from multiple sources to the target database, thereat performing a transformation process to structure the data. When the data is loaded into the target system, it can be queried and analysed as a simple Power BI example demonstrated.

The pipeline is flexible, meaning that making changes in the architecture of a pipeline should be easy to implement. Also, other sources can be added to the existing architecture with minimal changes. Furthermore, additional transformations can be easily implemented in existing pipelines if needed (Šandor and Bagić Babac, 2023).

However, enormous amounts of data require adjustments to the architecture in terms of tools and technologies used. For example, instead of Cloud Scheduler, Cloud Composer based on Apache Airflow should be used for orchestrating the data collection in a more effective and automated manner. Also, for more advanced manipulations of the data, Apache Spark can be used. Apache Spark can also be used for processing the data after the loading process.

5 Conclusions

Our paper provides practical solutions to common challenges in the development of data pipelines and demonstrates the effectiveness of using Google Cloud services for this purpose. Furthermore, we showcase the importance of flexibility in pipeline design to ensure adaptability to evolving data processing needs. To overcome the challenges which, come with big data, new technologies are being developed every day. Every new technology is trying to overcome the limitations of the previous one. It started with Hadoop, which was followed by Apache Spark, Apache Storm, and Apache Flink. A very important role in big data processing has cloud computing. The cloud offers users on-demand computing resources and enables building secure, scalable, efficient, and cost-effective big data processing systems.

The last part of this paper consisted of building an ETL pipeline taking into consideration all the challenges of big data processing and the best technologies for each use case, which were described in the paper. The developed data pipeline scrapes data from two sources on an hourly or daily basis. Data is then transformed to fit the model of the storage. Finally, the gathered data is stored, and an example of a report is made. For building a pipeline some state-of-the-art technologies were used, such as Google Cloud Platform and its services, Apache Beam and Power BI. The challenges such as data duplication needed to be tackled as well.

Our proposed data pipeline for efficiently handling big data has several potential applications. One such application is in the field of healthcare, where real-time processing of patient data can help doctors make more informed decisions (Liu et al., 2018). Another application is in the field of finance, where real-time processing of market data can help traders make more profitable trades (Chen et al., 2014). Our proposed solution can also be applied in the field of marketing, where real-time processing of customer data can help businesses make more targeted and effective marketing campaigns (Katal et al., 2013).

Big data offers numerous opportunities, but there are multiple challenges occurring along. For that reason, it is necessary to have a good knowledge of the technologies, methods, and tools and direct it properly into overcoming these challenges to extract valuable information from the data. Yielded information can then be used for making decisions, which can result in economic growth, enhancement of the education system, the advancement of science, improvement of health care and business, and many more.

For future work, there are several areas where our proposed solution could be extended and improved. One area is to investigate the performance of our solution on larger datasets and in more complex scenarios. Another area of future work is to evaluate the scalability of our proposed solution, particularly in cases where the volume of data increases rapidly. Additionally, it would be interesting to explore the use of machine learning techniques to further optimise the performance of our proposed solution (Puh and Bagić Babac, 2023).

Furthermore, our proposed solution has several potential applications in various fields that can be further developed and implemented to help businesses and organisations make more informed decisions based on the vast amount of data available to them (Brzić et al., 2023). Therefore, future work can explore the use of our solution in these different fields to assess its effectiveness and identify any additional challenges that may arise.

References

Abualkishik, A.Z. (2019) 'Hadoop and big data challenges', *Journal of Theoretical and Applied Information Technology*, Vol. 97, No. 12, pp.3488–3500.

Aion, M.K. (2016) 'Processing and analysis of enterprise ready big data in real-time', *Students Conference of Informatics, Electronics and Vision 2016*, University of Dhaka, Bangladesh.

Alguliyev, R., Gasimova, R. and Abbaslı, R. (2017) 'The obstacles in big data process', *International Journal of Modern Education and Computer Science*, Vol. 3, No. 3, pp.28–35, DOI: 10.5815/ijmecs.2017.03.04.

Bagić Babac, M. (2022) 'Emotion analysis of user reactions to online news', *Information Discovery and Delivery*, Vol. 51, No. 2, pp.179–193, https://doi.org/10.1108/IDD-04-2022-0027.

Bagić Babac, M. and Podobnik, V. (2016) 'A sentiment analysis of who participates, how and why, at social media sports websites: how differently men and women write about football', *Online Information Review*, Vol. 40, No. 6, pp.814–833, https://doi.org/10.1108/ITP-08-2016-0200.

Bezerra, A., Greati, V., Ribeiro, V., Silva, I. and Guedes, L.A. (2019) 'An industrial big data processing engine', *Anais do 14º Simpósio Brasileiro de Automação Inteligente*, Vol. 1, p.108372.

Blagaić, S. and Bagić Babac, M. (2022) 'Application for data migration with complete data integrity', *International Journal of System of Systems Engineering*, Vol. 12, No. 4, pp.405–432.

Brzić, B., Botički, I. and Bagić Babac, M. (2023) 'Detecting deception using natural language processing and machine learning in datasets on COVID-19 and climate change', *Algorithms*, Vol. 16, No. 221, https://doi.org/10.3390/a16050221.

Chambers, B. and Matei, Z. (2018) *Spark: The Definitive Guide*, O'Reilly Media, Inc., Beijing.

Chen, M., Mao, S. and Liu, Y. (2014) 'Big data: a survey', *Mobile Networks and Applications*, Vol. 19, No. 2, pp.171–209.

Cvitanović, I. and Bagić Babac, M. (2022) 'Deep learning with self-attention mechanism for fake news detection', in Lahby, M., Pathan, A.S.K., Maleh, Y. and Yafooz, W.M.S. (Eds.): *Combating Fake News with Computational Intelligence Techniques*, pp.205–229, Springer, Switzerland.

Domitran, S. and Bagić Babac, M. (2021) 'The use of deep reinforcement learning for flying a drone', *Journal of Information Science and Engineering*, Vol. 37, No. 5, pp.1165–1176, doi:10.6688/JISE.202109_37(5).0012.

Gartner, Inc. and/or its affiliates. (n.d.) *Definition of Big Data – Gartner Information Technology Glossary*, Gartner [online] https://www.gartner.com/en/information-technology/glossary/big-data (accessed 11 June 2022).

Hussien, A. (2020) 'Fifty-Six big data V's characteristics and proposed strategies to overcome security and privacy challenges (BD2)', *Journal of Information Security*, Vol. 11, No. 4, pp.304–328.

Jiang, L., Liu, L., Yao, J. and Shi, L. (2020) 'A hybrid recommendation model in social media based on deep emotion analysis and multi-source view fusion', *Journal of Cloud Computing*, Vol. 9, No. 57, https://doi.org/10.1186/s13677-020-00199-2.

Katal, A., Wazid, M. and Goudar, R.H. (2013) 'Big Data: a review', *International Journal of Computer Applications*, Vol. 74, No. 10, pp.1–7.

Khan, Z. (2021) 'Handling duplicate data in streaming pipeline using pub/sub & dataflow', *Google Cloud Blog* [online] https://cloud.google.com/blog/products/data-analytics/handling-duplicate-data-in-streaming-pipeline-using-pubsub-dataflow (accessed 25 June 2022).

Kitchin, R. (2014) 'Big data, new epistemologies and paradigm shifts', *Big Data & Society*, Vol. 1, No. 1, p.2053951714528481.

Lazer, D., Kennedy, R., King, G. and Vespignani, A. (2014) 'The parable of Google Flu: traps in big data analysis', *Science*, Vol. 343, No. 6176, pp.1203–1205.

Lipovac, I. and Bagić Babac, M. (2021) 'Content analysis of job advertisements for identifying employability skills', *Interdisciplinary Description of Complex Systems*, Vol. 19, No. 4, pp.509–523, doi:10.7906/indecs.19.4.5.

Liu, W., Fan, W., Li, P. and Li, L. (2018) 'Survey of big data platform based on cloud computing container technology', in Barolli, L. and Terzo, O. (Eds.): *Complex, Intelligent, and Software Intensive Systems. CISIS 2017. Advances in Intelligent Systems and Computing*, Vol. 611, Springer, Cham, https://doi.org/10.1007/978-3-319-61566-0_90.

Liu, X. and Iftikhar, N. (2015) 'An ETL optimization framework using partitioning and parallelization', *30th Annual ACM Symposium on Applied Computing*, ACM, Salamanca, Spain.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A. H. (2011) *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, McKinsey Global Institute.

Marijić, A. and Bagić Babac, M. (2023) 'Predicting song genre with deep learning', *Global Knowledge, Memory and Communication*, https://doi.org/10.1108/GKMC-08-2022-0187.

Mayer-Schönberger, V. and Cukier, K. (2013) *Big Data: A Revolution that will Transform How We Live, Work, and Think*, Houghton Mifflin Harcourt, Washington, DC.

McAfee, A. and Brynjolfsson, E. (2012) 'Big data: the management revolution', *Harvard Business Review*, Vol. 90, No. 10, pp.60–68.

Poch Alonso, R. and Bagić Babac, M. (2022) 'Machine learning approach to predicting a basketball game outcome', *International Journal of Data Science*, Vol. 7, No. 1, https://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijds.

Puh, K. and Bagić Babac, M. (2022) 'Predicting sentiment and rating of tourist reviews using machine learning', *Journal of Hospitality and Tourism Insights*, https://doi.org/10.1108/JHTI-02-2022-0078.

Puh, K. and Bagić Babac, M. (2023) 'Predicting stock market using natural language processing', *American Journal of Business*, Vol. 38, No. 2, pp.41–61, https://doi.org/10.1108/AJB-08-2022-0124.

Rehman, M.H., Yaqoob, I., Salah, K., Imran, M.A., Jayaraman, P.P. and Perera, C. (2019) 'The role of big data analytics in industrial internet of things', *Future Gener. Comput. Syst.*, Vol. 99, pp.247–259, https://doi.org/10.1016/j.future.2019.04.020.

Roukh, A., Nolack Fote, F., Mahmoudi, S. and Saïd, M. (2020) 'Big data processing architecture for smart farming', *Procedia Computer Science*, Vol. 177, No. 1, pp.78–85.

Šandor, D. and Bagić Babac, M. (2023) 'Designing scalable event-driven systems with message-oriented architecture', *Distributed Intelligent Circuits and Systems*, World Scientific (to appear).

Segaran, T. and Hammerbacher, J. (2009) *Beautiful Data: The Stories behind Elegant Data Solutions*, O'Reilly Media, Inc., Beijing.

Shakhovska, N. (2017) 'The method of big data processing', *2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv, Ukraine, Vol. 1, pp.122–126.

Sharma, D., Pabby, G. and Kumar (2017) 'Challenges involved in big data processing & methods to solve big data processing problems', *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, Vol. 5, No. 8, pp.841–847.

Sharma, R. (2018) *Security Issues of Big Data Hadoop*, GRIN Verlag, Munich.

Simaković, M.N., Cica, Z.G. and Masnikosa, I.B. (2021) 'Big Data architecture for mobile network operators', *15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pp.283–286.

Singh, L., Padden, C., Davis-Kean, P., David, R., Marwadi, V., Ren, Y. and Vanarsdall, R. (2021) *Text Analytic Research Portals: Supporting Large-Scale Social Science Research*, pp.6020–6022, IEEE, Orlando, FL, USA.

Stephenson, D. (2018) *Big Data Demystified: How to Use Big Data, Data Science and Ai to Make Better Business Decisions and Gain Competitive Advantage*, Pearson Education Limited, Harlow, England.

Suleykin, A. and Panfilov, P. (2019) 'Implementing big data processing workflows using open source technologies', *Proceedings of the 30th DAAAM International Symposium*.

Sulova, S. (2021) 'Big data processing in the logistics industry', *Economics and Computer Science*, No. 1, pp.6–19, Publishing house 'Knowledge and business' Varna.

Thabet, N. and Soomro, T. (2015) 'Big data challenges', *Journal of Computer Engineering & Information Technology*, Vol. 4, No. 3, DOI: http://dx.doi.org/10.4172/2324-9307.1000135.

Zaman, F.U., Hafeez, A. and Owais, M. (2021) 'Performance evaluation of Amazon's, Google's, and Microsoft's serverless functions: a comparative study', *International Journal of Scientific & Technology Research*, Vol. 10, No. 4, pp.189–192.

Zhang, H., Lee, S., Lu, Y., Yu, X. and Lu, H. (2023) 'A survey on big data technologies and their applications to the metaverse: past, current and future', *Mathematics*, Vol. 11, No. 96, https://doi.org/10.3390/math11010096.

Zhelev, S. and Rozeva, A. (2017) 'Big data processing in the cloud – challenges and platforms', *AIP Conference Proceedings*.

Notes

[1] https://hadoop.apache.org/.

[2] https://kafka.apache.org/.

[3] https://hadoop.apache.org/.

[4] https://www.macrometa.com/event-stream-processing/spark-vs-flink.

[5] https://airflow.apache.org/.

[6] https://www.postgresql.org/.

[7] https://cloud.google.com/.

[8] https://cloud.google.com/dataproc.

[9] https://datafloq.com/read/3vs-sufficient-describe-big-data/166.

[10] https://www.qlik.com/us/data-integration/data-pipeline.

[11] https://praw.readthedocs.io/en/stable/.

[12] https://newspaper.readthedocs.io/en/latest/.

[13] https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview.