Social Network Analysis and Mining (2022) 12:172
https://doi.org/10.1007/s13278-022-00996-4

ORIGINAL ARTICLE

A novel framework for remote management of social media big data analytics

Ahmad M. Al-Shomar[1] · **Muhammad Al-Qurish**[2] · **Wajdi Aljedaani**[3]

Received: 17 January 2022 / Revised: 29 October 2022 / Accepted: 1 November 2022 / Published online: 1 December 2022

Abstract
With the rapid expansion of social media users and the ever-increasing data exchange between them, the era of big data has arrived. Integration of big data generates enormous benefits, making it a hotspot for research. However, big data demonstrates the heterogeneity brought on by multiple data sources. Big data integration is constrained by multi-source heterogeneous data. Moreover, the rise in the volume of social media data is affecting the efficiency of data integration. This study is concerned with developing a novel framework for data integration system that can manage the heterogeneity of massive social media data. The framework is comprised of four layers: data source layer, application layer, resource layer, and visualization layer. The framework establishes correlations between data stored in distributed data sources. We used RESTful APIs to offer end-users with reliable and effective web-based access to data using unique queries. The framework was evaluated based on firsthand impressions of test users, who answered a standardized set of questions after testing real-world inputs.

Keywords Data analytics · Social media · Cloud solution · Dynamic integration

1 Introduction

The five Vs of big data-velocity, volume, value, variety, and veracity-define social networks (Abkenar et al. 2021). Big data generated by social media networks and their related analytics have attracted the attention of researchers with the development of technology. Presently, data are identified in unprecedented proportions in a variety of environments, increasing every 18 months as a consequence of various forms of databases such as databases obtained from different social media networks (Rossi and Hirama 2022). Big data is heterogeneous and complex in nature. Therefore, straightforward approaches are ineffective when trying to process and store big data.A primary issue is the diversity of data types and data inconsistency, redundancy, and incompatibility. Consequently, there is a perceived need to devise a reliable system that can retrieve, organize, and process information efficiently and safely.

The development of cloud computing has caused a divergence in how software and computers interact. Rather than having to install them locally, cloud computing enables the activation and accessibility of apps in the cloud (Alqarni 2021). Nevertheless, it is not surprising that user interest in cloud storage has surged. Microsoft, Google, Amazon, and others' services, which provide online storage, have sparked a gold rush following their arrival in 2012 (Ahuja et al. 2012). In a similar vein, the research community is constantly working to improve the effectiveness and security of cloud computing.

Data integration is the process of compiling data from multiple sources into a single, cohesive dataset. Researchers have been developing data integration tools (Kancharala 2021; Nie et al. 2021; VandanaKolisetty and Rajput 2021; Jung and Chung 2021). Bettio et al. (2021) developed MOMIS to integrate clinical data to visualize patient's natural, molecular, and clinical history. GNN-DDI was designed to integrate drug information from several systems by developing an attributed heterogeneous network

✉ Ahmad M. Al-Shomar
a.alshomar@uoh.edu.sa

Muhammad Al-Qurish
mualqurishi@elm.sa

Wajdi Aljedaani
wajdialjedaani@my.unt.edu

[1] University of Hail, Hail, Saudi Arabia

[2] Research and Innovation Division, Elm Company, Riyadh, Saudi Arabia

[3] University of North Texas, Denton, USA

(Al_Rabeah and Lakizadeh 2022). However, because the volume of data continues to rise dramatically in a relatively short period of time, the source data is oftentimes difficult to integrate constraining the effectiveness of data integration tools (Kalayci et al. 2021). Hilali et al. (2022) modified the ETL (Extraction, Load, and Transformation) process to handle the semantic heterogeneity of big data but lacked compatibility with NoSQL databases. For this, Our framework would help people to integrate the data to better use and manage the data from distributed data sources. Also, it is more valuable for businesses at a low cost. However, the traditional data integration framework does not determine clear integration standards as well as a higher cost to achieve effective data gathering, cleaning and integration.

The objective of this research is real-time consolidation of data from several social media networks into a distributed data source. In spite of a wider spectrum of data types and domains, the ultimate goal is to consistently offer and enable user access to data while taking into account commercial and application requirements. In order to accomplish this, we propose a generic framework for big data integration on a cloud environment for a swift, reliable, and safe access to social network data. The data from social media is collected using multiple APIs (Application Programming Interfaces). The proposed framework offers an interface for end-users to access the filtered data in a readable format. The main contributions of our study are listed below:

- Design and develop an integration framework to collect big data from distributed sources with different formats.
- Provide a unified format of the output data to be processed in cloud computing platforms.
- Propose a comprehensive framework and provide an interface for filtering data gathered from social media sites with the aid of big data integration.
- A replication package of our developed framework for extension purposes (https://github.com/AlShomar/AlShomar-Big-Data-Integration-Framework).

The rest of the paper is organized as follows: Sect. 2 discusses the related work. Section 3 presents the proposed framework, and discusses all the layers and algorithms used. In Sect. 4 illustrates the performance evaluation of the designed algorithms. Finally, Sect. 5 concludes the paper and highlights the direction of future work.

2 Related work

This research aims to address the lack of a standardized data integration framework for combining a centrally managed database with different standalone social media data sources. Integrating data from heterogeneous sources into one database has been a challenge for many researchers (Fillinger et al. 2019). Big data integration frameworks present four main issues including big data transformation, storage, and retrieval. Table 1 summarizes the relevant studies discussed in this section.

2.1 Big data integration

Several big data integration tools such as ROHDIP (Shehab et al. 2016), BINARY (Eftekhari et al. 2016), MOMIS (Bettio et al. 2021), GNN-DDI (Al_Rabeah and Lakizadeh 2022) etc, have been developed in recent years. Graph-based (Kancharala 2021), hybrid hierarchy architecture-based (Nie et al. 2021), data integration techniques have also been developed. Similarly, Probabilistic Semantic Association (PSA) was introduced to integrate big data by generating feature patterns for the data sources (VandanaKolisetty and Rajput 2021). Cluster-based data integration model was proposed by Jung and Chung (2021). Akinyemi et al. (2020) presented a framework that processes and integrates plant data which can help mitigate decommissioning costs and reuse decommissioned items. On the other hand, (Fletcher et al. 2019) employed weighted joint likelihoods in their data integration model as a mean to highlight data sources according to various criteria (e.g sample size).

The notion of big data integration in the cloud blends data manipulation technologies and cloud computing in a new generation of data analytics platforms (Kune et al. 2016; Manekar and Pradeepini 2017). Users today require new big data integration cloud services, such as data collection from many sources via cloud-deployed APIs.

2.2 Big data transformation

Integrating high-quality data into the cloud is not sufficient rather, data transformation is required to filter, combine, and modify or reformat data types (Dey and Pandit 2020). Li et al. (2021) used the bilinear data transformation method to map angular wind data to time series. Kim et al. (2021) proposed a data transformation architecture based on machine learning techniques. Similarly, a framework based on R programming language to transform SQL data to NoSQL format was proposed by Hasan et al. (2021). Vendor lock-in is one of the major challenges in big data integration that necessitates big data transformation. Ahmed et al. (2021) deployed the k-nearest neighbor (KNN) imputation method and Kaplan-Meier weights to deal with the transformation of sensor data. Arslan et al. (2019) developed a web-based software to distribute datasets by applying mathematical data transformation by computing the Pearson P test statistic.

Table 1 Summary of related work

Research area	Study	Year	Purpose	Tool
	Shehab et al. (2016)	2016	Providing uniform access to heterogeneous data sources	ROHDIP
	Eftekhari et al. (2016)	2016	Framework that supports integration, visualization & Ad-hoc querying	BINARY
	Fletcher et al. (2019)	2019	Integrating data from sources with varied sampling procedures	eBird
	Akinyemi et al. (2020)	2020	Integrated data for timely evaluation of decommissioned items	N/A
	Bettio et al. (2021)	2021	Integration of patient/disease information for clinical research	MOMIS
	Kancharala (2021)	2021	Semantic relations are discovered between integrated data items	N/A
	Nie et al. (2021)	2021	Classification of heterogeneous integrated data based on their theme	N/A
	Jung and Chung (2021)	2021	Social mining-based data integration improves health-risk prediction	N/A
Big data Integration	Al_Rabeah and Lakizadeh (2022)	2022	Prediction of drug-drug associated events using heterogeneous networks-based data integration	GNN-DDI
	Arslan et al. (2019)	2019	Normal distribution of dataset	Lambert W, arcsinh, Box-cox, Logarith--mic, Yeo-Johnson, and Square root
	Li et al. (2021)	2021	Transforming angular wind data to time-series data	bilinear Transformation
	Kim et al. (2021)	2021	Transform global waves to nearshore waves	ANN + GMDH
	Hasan et al. (2021)	2021	Transform several SQL databases to NoSQL	
Big data Transformation	Ahmed et al. (2021)	2021	Deal with censored data	KNN imputation and Kaplan–Meier Weights
	Shi et al. (2020)	2020	Provide reliable and systematic ad-hoc analysis system	Hadoop
	Honar Pajooh et al. (2021)	2021	Store IoT data with minimum latency abd cost	HLF
	Viswanath and Krishna (2021)	2021	Secure storage and retrieval of big data	
	Saenko and Kotenko (2022)	2022	Improve replica distribution across storage nodes	HDFS
Data Storage and Retrieval	Arer et al. (2022)	2022	Overcoming storage overhead and scalability issues	IPFS (storage) elasticsearch (retrieval)
	Ye et al. (2022)	2022	Improving retrieval efficiency of video data	Parallel Top-N

2.3 Big data storage and retrieval

Big data storage necessitates better storage functionalities. Saenko and Kotenko (2022) focused on providing efficient and resilient big data storage based on Hadoop Distributed File System (HDFS). Honar Pajooh et al. (2021) considered Hyperledger Fabric (HLF) platform with decentralized storage for IoT data. Authors enhanced data integrity by storing meta-data in off-chain big data systems. Shi et al. (2020) devised a Hadoop-based system using cloud storage to provide an efficient decision system. Data security is a significant issue that researchers are trying to solve regarding cloud storage. Viswanath and Krishna (2021) designed an encryption technique that was primarily responsible for securing the big data stored in a multi-cloud environment.

When retrieving large data, it is difficult to fully meet the expectations of end-users because traditional retrieval methods are prevalently time-consuming and take little consideration of the multi-source diverse attributes of big data. Arer et al. (2022), used IPFS(Inter-Planetary File System) for big data storage and elasticsearch for efficient data retrieval. Another study (Ye et al. 2022) used a parallel top-N algorithm to summarize and swiftly retrieve the matching semantic features of video data in big data.

3 Proposed approach

We propose a big data integration framework on the cloud as shown in Fig. 1. The proposed framework consists of the data source layer, application layer, resource layer, and visualization layer.

3.1 Data source layer

The data source layer provides big data from distributed data sources and connects directly to the application layer. This layer contains three social networks, namely, Twitter, YouTube, and Facebook. It is dependent on the intent of the application layer.

3.1.1 Twitter social media

Twitter as a social network. Since 2006, Twitter has been widely used among Internet users Al-Qurishi et al. (2018), resulting in some related literature on Twitter to understand microblogging usage and communities better. Twitter users can classify their posts into four categories: daily chatter, conversations, sharing information or links, and reporting news. The role of Twitter users can be classified into three classes:

- Broadcasters, those who have a huge number of followers;
- Acquaintances, those who have approximately the same number of followers and following; and
- Miscreants and evangelists follow a huge number of users but have only a few followers.

The use of Twitter goes beyond personal use, and it can be convenient. Businesses consider it a channel to increase awareness about their products, create business opportunities, maintain customer loyalty, host marketing campaigns, improve reputation, predict trends, and recruit new talents Al-Qurishi et al. (2018).

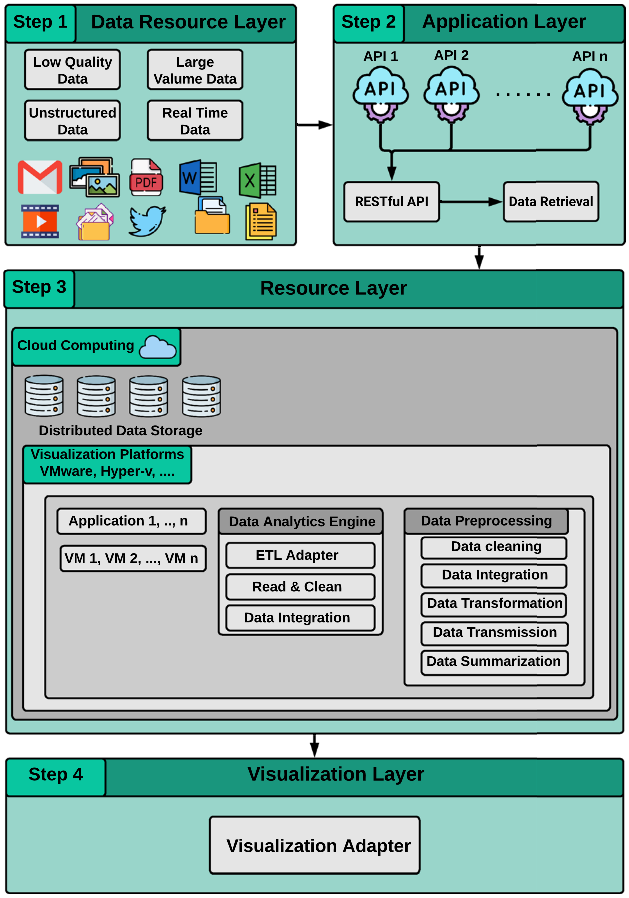

Fig. 1 Architecture of the proposed framework

3.1.2 YouTube social media

YouTube as a social network. Since 2005, YouTube has been a video-sharing website, which started as a media tool and became a marketing communication tool. It is a rich tool that contains multiple mechanisms, such as trending, subscribers, and a list of related videos, which could affect how a video is published, thereby impacting its popularity. Users of YouTube across the globe can upload free video content and generate billions of views every day. These millions of users can significantly affect the reputation of an organization or a person. The use of YouTube goes beyond personal use, and it can run ongoing information about new services or products. Moreover, many factors have facilitated the growth of its use, including ease of uploading videos, accessing commercial content, education, and broadcasting networks.

3.1.3 Facebook social media

Facebook is a social networking services website launched in 2004, allowing users to create profiles, send messages, and keep in touch with friends. The Facebook website represents a huge potential market for social media efforts. Facebook users can be categorized by their use as sharing status, social connection, sharing identities, and browsing the social network. The use of Facebook fulfills two needs: belonging and self-presentation. The belonging need allows users to learn about others and communicate with them, which is a significant motivator. The self-presentation need includes creating user profiles and posting images and wall content. It can be used for social searching, finding out information about offline users, and social browsing, which is used to develop new connections for offline interaction.

3.2 Application layer

This layer provides a link between the resource layer and the data source layer as shown in Fig. 2. It comprises a set of RESTful APIs and a data retrieval algorithm. The function of RESTful APIs is to collect big data from social media data sources and transfer the data to the data retrieval algorithm.

3.2.1 RESTful APIs

REST architecture stands for Representational State Transfer (REST) and is used to deploy large-scale distributed systems based on the client-server model, which can exchange data between applications or systems. In REST, everything is a resource, and these resources can be accessed by the application program interface using the HTTP protocol. REST APIs are vital to pulling data from distributed data sources based on end-user requests. In this paper, we have used to connect the data source layer and the application layer. Those APIs are from social media channels, which are used to collect and retrieve data from them. These data can be used to read, update, create and delete data types, as shown in Fig. 3.

3.2.2 Data retrieval algorithm

The data retrieval algorithm inserts the topic-driven by passing two parameters: the keyword and access token. More details of this algorithm are presented below in Algorithm 1.

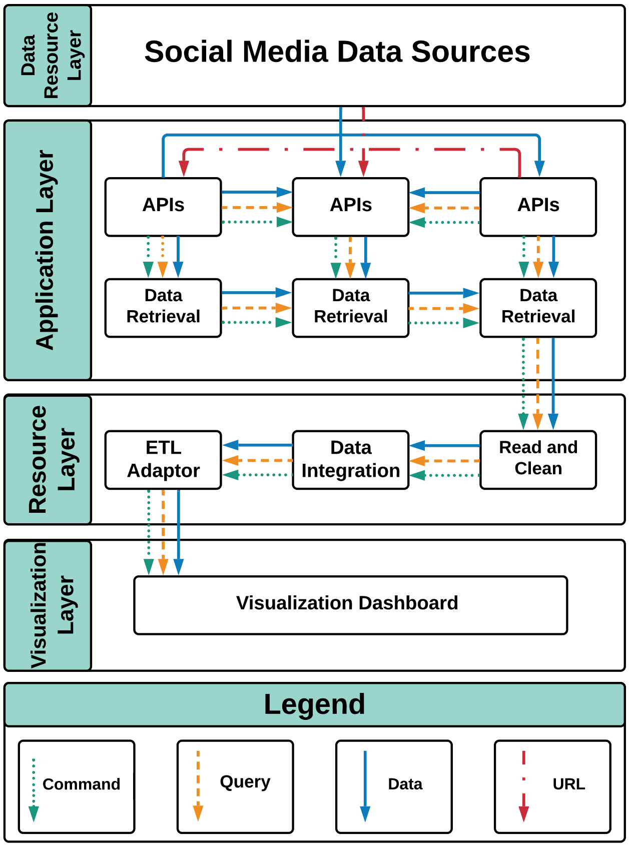

Fig. 3 API connectivity among the four layers of the framework

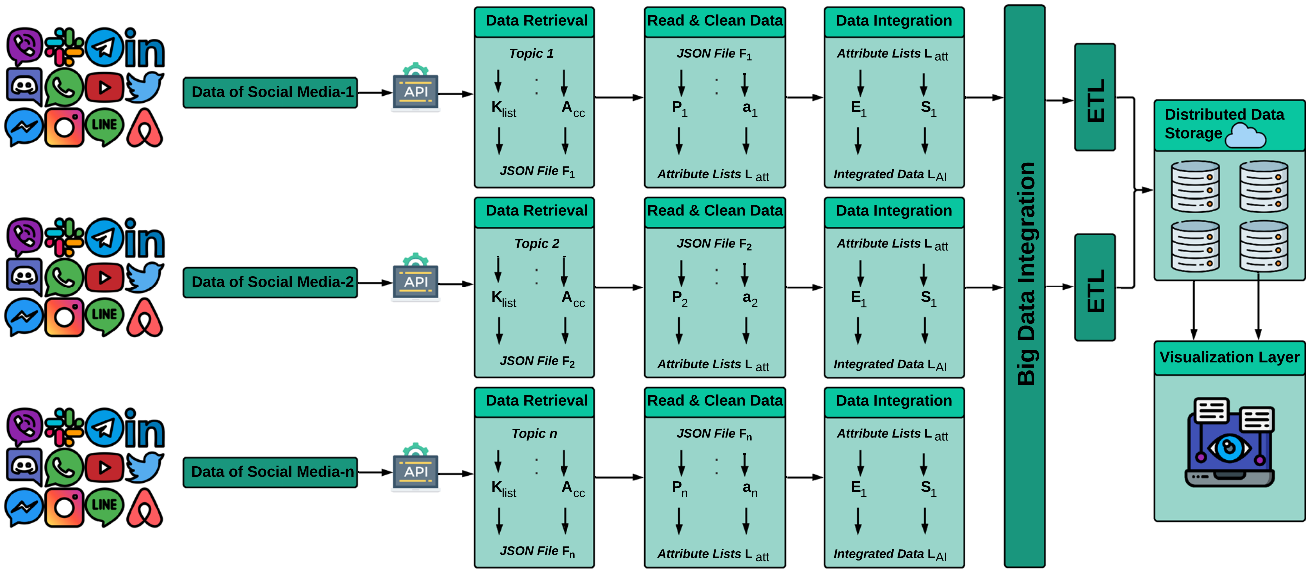

Fig. 2 A block diagram of the proposed framework

```
Algorithm 1: Data Retrieval
  Input: $Topic$
  Output: $F$, where $F$ is a file in JSON format
1  $K_{list} = 1, 2, 3, ..., n$;
2  GetDataService (API);
3  Initial parameter ($K_{list}, A_{cc}$);
4  format(queryURL);
5  $Q \leftarrow$ Add($queryURL$);
6  foreach $q \in Q$ do
7      if $q$ is valid then
8          $data \leftarrow$ responseResults($q$);
9          $F \leftarrow$ buildJSONFile($data$);
10     else
11         return empty;
12     end
13 end
14 return $F$;
```

Definition 1 (Data retrieval) *Topic* denotes a keyword-driven topic. K_{list} denotes a keyword ID list. API denotes an application program interface. A_{cc} denotes the access token of the social network. Q stores the URL query. q denotes an element in Q. F denotes the file in JSON format. Algorithm 1 acquires the data object with a given API using the function GetDataService (API), which is an interface to collect big data from a given social media service. Hence, it initializes two parameters: keyword-driven K_{list} and access token A_{cc}. Then, it formats the queryURL using the function FORMAT (queryURL) before adding the query to Q. In steps 6–10, check whether the URL is valid or not. If the URL is not valid, it returns the result of an empty file. Otherwise, it builds a JSON file with the retrieved big data in F notation. Finally, the algorithm returns the file in JSON format represented in F, which includes the collected big data.

3.3 Resource layer

This layer comprises the read and clean data algorithm, data integration algorithm, distributed storage, ETL adapter, and cloud computing environment, which support the implementation of remote access, servers, virtual machines, and hardware and software resources. The resource layer satisfies the requirement of distributed big data storage and management of all host servers.

3.3.1 Read and clean data algorithm

The read and clean data algorithm inserts a file in JSON format and outputs lists of social media data attributes. For more details, please see Algorithm 2 below.

```
Algorithm 2: Read and Clean Data
  Input: $F$, where $F$ are data of a JSON file
  Output: {$L_{att}$: $A_{ut}$, $A_{tw}$, $A_{fb}$} are lists of
          extracted data attributes
$F \leftarrow$ read(JSONFile);
foreach $a \in F$ do
    $attibuteList \leftarrow$ extractAtt($a$);
    $pa \leftarrow$ parse($attibuteList$);
    $L_{att} \leftarrow$ add($pa$);
end
return $L_{att}$;
```

Definition 2 (Read and Clean Data). F denotes a JSON file. L_{att} denotes the list of attributes that come from social media sites. a is an element in JSON file F. A_{ut} represents the attributes of YouTube; A_{tw} denotes the attributes of Twitter, and A_{fb} contains the attributes of Facebook. *Att* denotes the attributes. *pa* denotes storing the attributes after being extracted and parsed. Algorithm 2 reads the file in JSON format with the given function F. Then, it cleans the data by extracting the attributes that denote extract *Att*(a) of the JSON file F, which comes from different social media sources. The next step will be parsing the big data attributes using the Parse *Att*(a) based on social media networks. After the attributes are extracted and parsed, it will store the attributes of A_{ut}, A_{tw}, and A_{fb} into L_{att} using the function $L_{att} \leftarrow$ Add(pa). Finally, the algorithm returns the lists of attributes including A_{ut}, A_{tw}, and A_{fb} into L_{att}.

3.3.2 Data integration algorithm

The big data integration algorithm takes the list of extracted data attributes in the form of Twitter, YouTube, and Facebook. Each element in the social media data is selected based on these attributes and is stored in the list of integrated data sources. Then, the output will be the integration of multiple data types and various formats of the integrated data sources. More details of this algorithm are presented in the Algorithm 3.

```
Algorithm 3: Data Integration
  Input: $L_{att}$: List of extracted data attributed in the
         form $A_{ut}$, $A_{tw}$, and $A_{fb}$
  Output: $L_{AI}$: List of integrated big data
1  foreach $E_{ut} \in A_{ut}$ do
2      $S_{ut} \leftarrow$ select($E_{ut}$);
3      $L_{AI} \leftarrow$ add($S_{ut}$);
4  end
5  foreach $E_{tw} \in A_{tw}$ do
6      $S_{ut} \leftarrow$ select($E_{tw}$);
7      $L_{AI} \leftarrow$ add($S_{ut}$);
8  end
9  foreach $E_{tb} \in A_{fb}$ do
10     $S_{ut} \leftarrow$ select($E_{fb}$);
11     $L_{AI} \leftarrow$ add($S_{ut}$);
12 end
13 return $L_{AI}$
```

Definition 3 (Big Data Integration). L_{att} denotes the list of attributes for YouTube, Twitter, and Facebook. L_{AI} denotes the list of integrated data sources. A_{ut} represents YouTube attributes; A_{tw} contains the Twitter attributes, and A_{fb} denotes the attributes of Facebook. E_{ut} denotes an element in YouTube. E_{tw}denotes an element in Twitter. E_{fb} denotes an element in Facebook. S_{ut} denotes the selected element. Algorithm 3 is divided into three parts based on the social networks. In the first part (steps 1–4), the element of YouTube E_{ut} is selected based on the attributes using Select (E_{ut}) and is stored in the selected element S_{ut}. L_{AI} derives the selected S_{ut} of the given function using the function $L_{AI} \leftarrow$ Add(S_{ut}). The second part of the algorithm (steps 5–8) describes the element of Twitter E_{tw}which is selected based on attributes using the function $S_{ut} \leftarrow$ Select(E_{tw}), and stores it in the selected element S_{ut}. L_{AI} derives the selected S_{ut} of the given function using Add (S_{ut}). Finally, in the third part of the algorithm (steps 9–12), the element of Facebook E_{fb} is selected based on the attributes using the Select (E_{fb}) function and is stored in the selected element S_{ut}. L_{AI} derives the selected S_{ut} of the given function $L_{AI} \leftarrow$ Add(S_{ut}). Hence, L_{AI} returns the list of integrated datasets.

3.3.3 ETL adapter

ETL stands for Extract-Transform-Load and covers a process of pulling out data from one source to another. The ETL adapter supports the transfer and reduction of the integrated big data into the distributed storage. The extract step has been used to retrieve the data from the source system in a way that does not affect the performance or response time. The next step will be transforming the data from the source to the target place using the same dimension so that it can be joined later. The use of the load step fulfills the loading correctly with little resources that allow us to stop any constraints before the loading step and enable them after the load is completed.

3.4 Visualization layer

After the integrated data is stored in a reliable data source, we need to search, view, and interact with these datasets using an analytics engine and RESTful search. These data can be accessed through RESTful APIs and uses the JavaScript Object Notation (JSON) schema to store data. Moreover, we provide the capabilities to visualize the data in various maps, tables, and charts on top of large volumes of data. Search engine queries in real-time can do this visualization of data. In addition, users can create and share dynamic dashboards that display any changes in search engine queries.

4 Performance evaluation

We designed three algorithms to integrate big data from different sources with different formats. The experiments aim to evaluate the performance of these algorithms. We believe that the performance of the designed algorithms needs to be optimized and tested in a single server instance before we consider scaling up and scaling out. These experiments compare the performance of the algorithms during the execution from different data sources such as Facebook, YouTube, and Twitter.

4.1 Experimental setup

The designed algorithms were implemented in Java 7, Apache Maven 3.5.0, and Redis 4.0.1. We used Elasticsearch 5.6.0 as storage and Apache Kafka as the stream processing platform. All of these tools were deployed on the same virtual cloud machine. This machine runs on the following operating system: Ubuntu 64-bit 17.4 with Intel Core i5 2.40 GHz CPU, 10 GB of RAM, and 250 GB of hard disk storage.

4.2 Experimental dataset

We employ in our experiments three datasets from popular social media sites (Twitter, YouTube, and Facebook). These datasets were collected randomly by the data retrieval algorithm using social media APIs, which consists of three queries. We applied the first twitter4j query on Twitter which is a %100 pure Java library for the Twitter API in order to retrieve the maximum number of tweets and replies and we were able to collect 122,432 data size/ms. For the second query, we applied access token query for YouTube channel which is send Web API requests with an access token included either in the HTTP Authorization header or as a POST parameter in order to collect the data. Hence, we collected 91,146 data sizes/ms from YouTube consisting of comments and replies. For the third query, we applied Redis query on Facebook since the keys can contain hashes and sorted sets and were able to retrieve comments and replies around 56,260 data size/ms. Finally, we integrated these datasets using data integration algorithms. Table 2 summarizes the datasets used to evaluate the three algorithms from distributed sources.

4.2.1 Data retrieval algorithm (Exp. 1)

This experiment aims to benchmark the execution time of the data retrieval algorithm from Twitter, YouTube, and Facebook. We employ the algorithm to collect the datasets from the social media APIs, which are composed of three subsidiary queries. The first query $Q1$ retrieves the maximum number of tweets and replies from Twitter and checks if the file is empty; otherwise, a JSON file F is built. The second query $Q2$ retrieves the maximum number of video comments and comment replies from YouTube and checks whether the file is empty; if not, a file in JSON format F is built. The third query $Q3$ retrieves the maximum number of Facebook post comments and Facebook comment replies and checks if the file is empty; if not, a JSON file F is built. The same function executes $Q1$, $Q2$, and $Q3$ and builds the JSON file F. Note that the slowness of execution rate for Twitter while running the code takes 10 to 20 Seconds because of the get_constent() function, unlike YouTube and Facebook.

4.2.2 Read and clean data algorithm (Exp. 2)

This experiment aims to benchmark the execution time of the read and clean data algorithm to read the JSON file F and clean the data. The algorithm has three functions: extracting, parsing, and adding the attributes of Twitter, YouTube, and Facebook. The Twitter attributes A_{tw} will be extracted and parsed from JSON file F for both tweets and replies. The YouTube attributes A_{ut} will be extracted and parsed from JSON file F for video comments and reply comments. For the Facebook attributes, A_{fb} will be extracted and parsed for post comments and replies to post comments. All of A_{ut}, A_{tw}, and A_{fb} are added to the lists of attributes L_{att}. The same function executes A_{ut}, A_{tw}, and A_{fb} and produces a list of social media attributes.

4.2.3 Data integration algorithm (Exp. 3)

The aim of this experiment is to benchmark the execution time of the data integration algorithm. This algorithm has two major functions: selecting social media attributes and adding these attributes to the list of integrated big data L_{AI}. Each element in social media such as Twitter E_{tw}, YouTube E_{ut}, and Facebook E_{fb} is being selected based on specified attributes such as Channel(), CommentId(), OwnerId(), ParentId(), PublishedTime(), and Type() for the main post as well as the replies. The selected elements will be added to the lists of integrated big data L_{AI}. Hence, the algorithm returns the integrated big data lists that come from different sources. The same function executes Twitter E_{tw}, YouTube E_{ut}, and Facebook E_{fb}.

4.3 Experimental results

In general, these results suggest that the algorithms can effectively retrieve, read, clean and integrate the data from distributed data sources. In this section, we present the results of the three experiments on data retrieval, read and clean data, and data integration for social media sites. the execution rate for both data retrieval and read and clean data algorithms were acceptable rates. Furthermore, the data integration algorithm was at a good rate.

4.3.1 Data retrieval algorithm

Exp. 1 shows that the execution rate of the data retrieval algorithm of $Q1$ performed on Twitter is 286.5 (data size/ms), $Q2$ performed on YouTube is 27.9 (data size/ms), and $Q3$ performed on Facebook is 30.7 (data size/ms). Hence, the execution rate of the data retrieval algorithm for

Table 2 Datasets used to evaluate the proposed algorithms

Social media sites	Datasets for data retrieval algorithm	Datasets for read and clean data algorithm	Datasets for data integration algorithm
Twitter	43040 data size/ms	40321 data size/ms	39071 data size/ms
YouTube	36022 data size/ms	32032 data size/ms	23092 data size/ms
Facebook	20950 data size/ms	18920 data size/ms	16390 data size/ms

YouTube is lower than that of Twitter and Facebook. The results of this experiment are shown in Figure 4.

4.3.2 Read and clean data algorithm

Exp. 2 shows that the execution rate of the read and clean data algorithm of tweets is 504.1 (data size/ms) and replies of tweets are 6.48 (data size/ms). For the comments on the YouTube video, it is 38.5 (data size/ms) while it is 3.89 (data size/ms) for the replies to video comments. Finally, for Facebook post comments, the execution rate is 13.9 (data size/ms) and 4.6 (data size/ms) for Facebook comment replies. Hence, the execution rate of the read and clean data algorithm for Facebook is lower than that of Twitter and YouTube. The results of this experiment are shown in Figure 5.

4.3.3 Data integration algorithm

Exp. 3 shows that the execution rate of the data integration algorithm of tweets integration is 461.9 (data size/ms) and 9.0 (data size/ms) for the replies. On the other hand, the rate is 37.8 (data size/ms) for YouTube video comments integration and 18.9 (data size/ms) for comment replies. Finally, the execution rate is 11.7 (data size/ms) for Facebook post comments and 6.3 (data size/ms) for replies to post comments. Hence, the execution rate of the data integration algorithm for Facebook is lower than that of Twitter and YouTube. The results of this experiment are shown in Fig. 6.

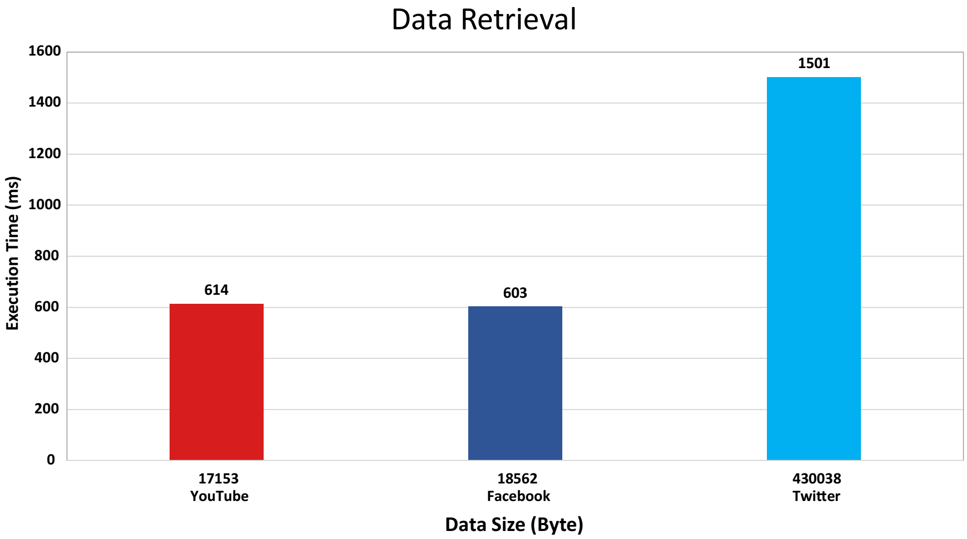

Fig. 4 Result of the data retrieval algorithm

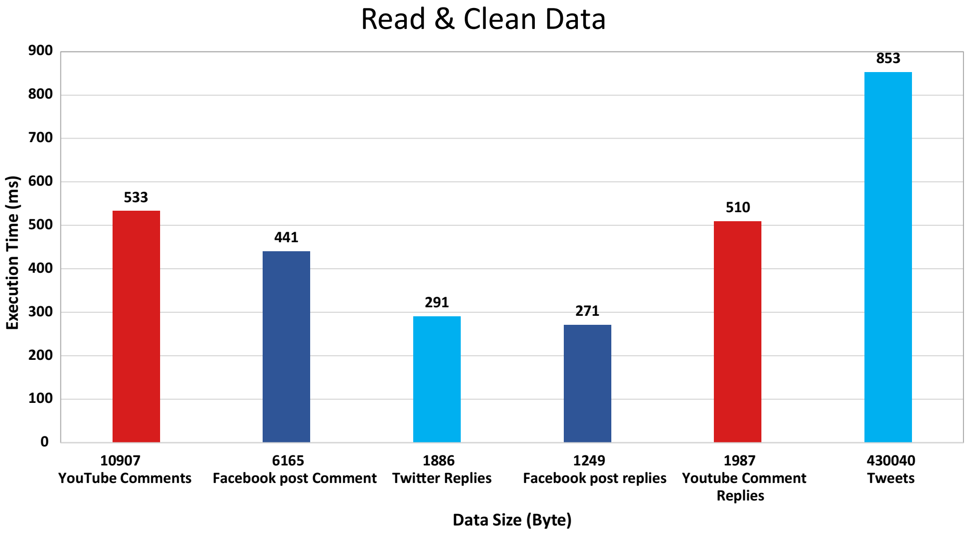

Fig. 5 Result of the data read and clean algorithm

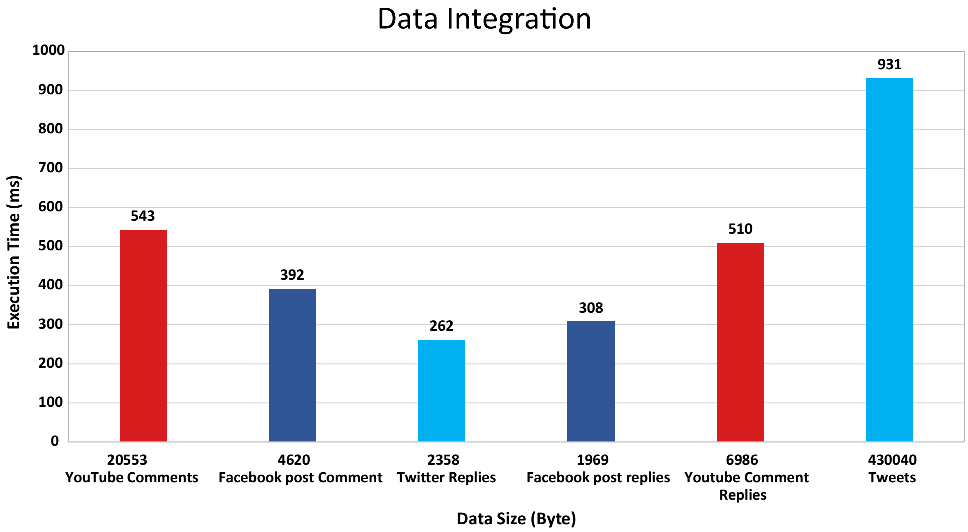

Fig. 6 Result of the data integration algorithm

4.4 Comparison with manual data integration

The proposed solution is compared to manual data integration based on the values of each tuple that are essential for extracting the correct information. Table 8 summarizes the comparison of manual data integration with proposed solution. The first issue is the integration type that will contain all the required data. The manual integration is going to be arduous because each line in the spreadsheet has to be manually populated with data. The proposed solution simplifies this process by gathering and integrating the data into distributed data sources using mapping techniques. It is going to eliminate any error that may be occurred. Another advantage of the proposed solution is that the scale of the data is too large compared to manual integration. This is useful where a huge amount of data are processed. Furthermore, the execution rate of the proposed solution is good while the execution rate of manual integration is questionable because some columns have to be manually ordered and may be forgotten or hidden if the user does not pay detailed attention. The use of manual data integration includes manually resolving meaning between terms to make sure that they belong to the same thing. The proposed solution on the other hand uses exact matching for considering the meaning and relationships between terms. Another issue with manual integration is that if the user does not pay attention to the rows of data, the empty cell may exist and the data is overlooked (Table 3).

Table 3 Comparison of manual data integration and proposed solution

S/N	Issues	Manual data integration	Proposed solution
1	Integration Type	Manual integration	Automatic integration
2	The scale of data	Large	Too large
3	Execution rate	Questionable	Good
4	Heterogeneity	Manual resolve	Automatic resolve
5	Missing values	Empty cell Exist	Null cell exist

4.5 Evaluating the big data integration framework

After completing the design of the framework, we invited participants to fill out a questionnaire to determine their cognitive behavior and evaluate the big data integration framework. To conduct the evaluation properly, we utilized the ARCS model invented by Keller (1983). ARCS stands for (A) attention, (R) relevance, (C) confidence, and (S) satisfaction. The model is generally used for evaluating how users interact with a system with the aid of user performance in an interactional framework. The calculated score of the ARCS model is 9 points based on the Likert scale, where 9 score is the highest and 1 is the lowest score Paas et al. (1994) (refer to Table 4 for the questionnaire based on the ARCS model). In our framework context, the attention items refer to the responses of the framework users about performing various queries and functions. The relevant items helped users to determine whether the results of the social media data are similar to real-life situations or not. The confidence items refer to the performance of the framework users in terms of facing any complexity or some difficulty concerning the framework. Finally, the satisfaction items measure the users' experience in integrating big data and how the results meet the users' expectations. To evaluate the quality of the above questions in terms of attention, relevance, confidence, and satisfaction, we applied some statistical procedures for each item. The statistical procedures are Cronbach's alpha, means, variance, and Pearson correlation coefficients.

Table 4 Questionnaire based on ARCS model

Attention
The web-based application for the framework was very interesting
The topic-driven in the search bar was really credible
The use of distributed social media sources was very good
The framework control from the different social media was exciting
The framework environment was motivating and was linked to real time
The practices of big data integration attributes caught my attention
The framework was very helpful and well-designed
Relevance
The framework details were related to my interests
The integrated social media data helped the self-monitoring
The topic-driven search was similar to daily social life
It was very helpful for self surveillance of social media
Confidence
The framework design was too difficult to understand
The results of the search bar were too complex
The final goal of the framework is so difficult to understand
Choosing the social media to be integrated inside the framework is not easy
The framework attributes were not enough and were not suitable for the end-user
Satisfaction
The framework helped the users to integrate social media data
I felt happy when I used the framework successfully
The integrated social media attributes exceeded my expectation
Comparing social media data inside the framework is good and adequate for the users

Table 5 Cronbach's alpha measurement

Cronbach's alpha	Internal consistency
$0.9 \leq \alpha$	Excellent
$0.8 \leq \alpha < 0.9$	Good
$0.7 \leq \alpha < 0.8$	Acceptable
$0.6 \leq \alpha < 0.7$	Questionable
$0.5 \leq \alpha < 0.6$	Poor
$\alpha < 0.5$	Unacceptable

4.5.1 Cronbach's coefficient alpha

Cronbach's coefficient is used to measure the scale of reliability in terms of the internal consistency between the observed and true scores Petri et al. (2017). To measure the internal consistency, you have to prove the scale of the question having one dimension. The Cronbach's coefficient has the following formula:

$$\alpha = \frac{N \cdot \bar{c}}{\bar{v} + (N - 1) \cdot \bar{c}} \tag{1}$$

Here, the number of items is denoted as N, $\bar{c}$ represents the average covariance among the items, and $\bar{v}$ is equal to the average variance. Based on the above formula, the raw value of Cronbach's coefficient must be acceptable or better to apply the internal consistency according to Table 5.

A single question was presented (Item) to each participant to check whether the 20 questions in terms of attention, relevance, confidence, and satisfaction have achieved internal consistency or not. We used a Statistical Software suite (SAS)(NoAuthor 2020) which helps to compute the internal consistency using Cronbach's coefficient α for all questions (items). After we imported the values into SAS, a reliability coefficient test was calculated on each item. to measure the internal consistency. Consequently, Cronbach's coefficient α is 0.83 for all items, which has an acceptable internal consistency.The result returns two coefficients: raw and standardized; the raw coefficient depends on the item correlation so that when the test is consistent, then the more robust the items are interrelated. The standardized reflects the item covariance and is used to measure the distribution of two variables covariance is used to measure the distributions of two variables. When the correlation coefficient is higher, the covariance is higher. Now, we will use a dataset that contains 20 variables imported from a questionnaire based on the ARCS model in Table 4 to measure internal consistency using Statistical Software suite (SAS)(NoAuthor 2020). The result shows that The alpha coefficient for the 20 items is 0.839, suggesting that the items have relatively high internal consistency. Note that in most social science research situations, Internal Consistency coefficient of 0.70 or higher is considered "acceptable" (see Tables 5, 6 and 7 below).

4.5.2 Variance

After the above demonstration, we calculated the variance of all the items to measure how far the variables are spread out from each other in the datasets by using the formula below:

$$\alpha^2 = \frac{\sum (X - \mu)^2}{N} \tag{2}$$

The variance α^2 is the sum of the squared distance of each item from μ divided by the number of items. After applying the above formula to the ARCS model through the SAS application, we obtain the outputs given in Table 8.

Delete "As overall results for attention, relevance, confidence, and satisfaction, we obtained Fig. 7 below."

Table 6 Results of Cronbach's coefficient α: Alpha values

20 Variables	Q1, Q2, Q3, ..., Q20
Cronbach Coefficient Alpha	
Variables	*Alpha*
Raw	0.839859
Standardized	0.841228

4.5.3 Pearson correlation coefficient

The correlation coefficient is used to measure the relationship among the variables, and its value is always between +1 and −1 (Puth et al. 2014). One of the most commonly used formulas for correlation is the Pearson correlation coefficient, which measures the linear relationship between datasets and shows how they are related to each other (refer to the formula below):

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \tag{3}$$

where N, $\sum xy$, $\sum x$, $\sum y$, $\sum x^2$, $\sum y^2$ represent the sum of item scores, the sum of xy paired scores, the sum of x scores, the sum of y scores, the sum of squared x scores, and the sum of squared y scores, respectively. Here, we apply the Pearson correlation coefficient for all ARCS models and extend how the standards are interrelated with 20 variables after applying the Pearson correlation coefficient, where -1 perfectly negative linear relationship, 0 being no correlation, and +1 perfectly positive linear relationship. See Table 9.

Table 7 Results of Cronbach's coefficient α: simple statistics

Simple statistics						
Variable	N	Mean	Std dev	Sum	Minimum	Maximum
Q1	5	3.6	2.07	18	2	7
Q2	5	5.2	3.76	26	1	9
Q3	5	6.2	3.03	31	3	9
Q4	5	4.2	3.63	21	1	9
Q5	5	6.2	3.11	31	2	9
Q6	5	2.4	1.67	12	1	5
Q7	5	5.2	2.48	26	3	9
Q8	5	3.6	3.20	18	1	9
Q9	5	5.8	3.42	29	1	9
Q10	5	3.6	3.40	18	1	9
Q11	5	5.2	3.11	26	2	9
Q12	5	4.4	2.30	22	2	7
Q13	5	4.8	3.42	24	2	9
Q14	5	5.0	2.34	25	3	9
Q15	5	6.6	1.51	33	5	9
Q16	5	4.8	2.77	24	1	8
Q17	5	3.6	3.20	18	1	8
Q18	5	7.0	2.91	35	2	9
Q19	5	4.4	1.51	22	2	6
Q20	5	4.0	3.39	20	1	9

Table 8 Results of the variance

Variable	Variance
Q1	4.3
Q2	14.2
Q3	9.2
Q4	13.2
Q5	9.7
Q6	2.8
Q7	6.2
Q8	10.3
Q9	11.7
Q10	10.3
Q11	9.7
Q12	5.3
Q13	11.7
Q14	5.5
Q15	2.3
Q16	7.7
Q17	10.3
Q18	8.5
Q19	2.3
Q20	11.5

We can obtain the scatter plot matrix among all items, i.e., attention, relevance, confidence, and satisfaction, as shown in Fig. 7.The scatter plots below in Fig. 7 shows that two distinct properties are the direction and strength of a correlation. The direction of the correlations gives a negative correlation corresponding to a decreasing relationship, while and a positive correlation corresponds to an increasing relationship. For the strength of a correlation can be assessed by taking values $0.1< | r |< 0.3$ as weak correlation, values like $0.3 < | r | < 0.5$ represent moderate correlation, and values like $.5 < | r |$ represent strong correlation. As shown in the scatter plot below, Attention and Satisfaction have positive and strong correlations While Confidence has a positive and moderate correlation. Finally, Relevance has a positive and weak correlation.

5 Conclusion and future work

With the massive growth of data, along with the availability of data manipulation and credible distributed data storage, many organizations no longer rely on conventional data processing to manipulate their datasets. They are moving toward big data models to combine datasets from multiple data sources that utilize various formats. Hence, having access to distributed big data sources and cloud computing technologies, and scalable big data storage is essential for integrating big data into the cloud.

Currently, big data paradigms have many limitations in the following aspects: they do not perform big data integration, and they have no reliable big data storage; they do not have a comprehensive view for query and data integration of multiple data types, as well as various formats of integrated data sources through cloud computing; hence, they only rely on the data types themselves and RDBMSs. In this research, we showed the proposed framework for social media big data integration on the cloud, which has the following properties:

- The framework has three algorithms to retrieve data from social media channels and facilitate reading big data through the data source layer and application layer.
- Big data are integrated from multiple data types and formats of distributed data sources on the resource layer using the data integration algorithm.
- The framework provides a web interface to formulate responses based on the user service demand.

The framework finds the correlation between the data stored in distributed data sources, supporting various formats of data types, providing access to huge datasets with the help of cloud computing technologies, and enabling queries and ubiquitous data access. The framework has four layers:

1. The data source layer, which is responsible for providing the dataset from distributed data sources.
2. The application layer with a data retrieval algorithm is used to retrieve big data from distributed data sources and to build files in JSON format.

Table 9 Pearson correlation coefficient results

Pearson correlation coefficients Prob > |r| under H0: Rho=0				
	Attention	Relevance	Confidence	Satisfaction
Attention	1.00000 35	0.40094 0.0798 20	0.22904 0.2708 25	0.23529 0.3180 20
Relevance	0.40094 0.0798 20	1.00000 20	−0.27634 0.2382 20	0.27126 0.2473 20
Confidence	0.22904 0.2708 25	−0.27634 0.2382 20	1.00000 25	0.30662 0.1885 20
Satisfaction	0.23529 0.3180 20	0.27126 0.2473 20	0.30662 0.1885 20	1.00000 20

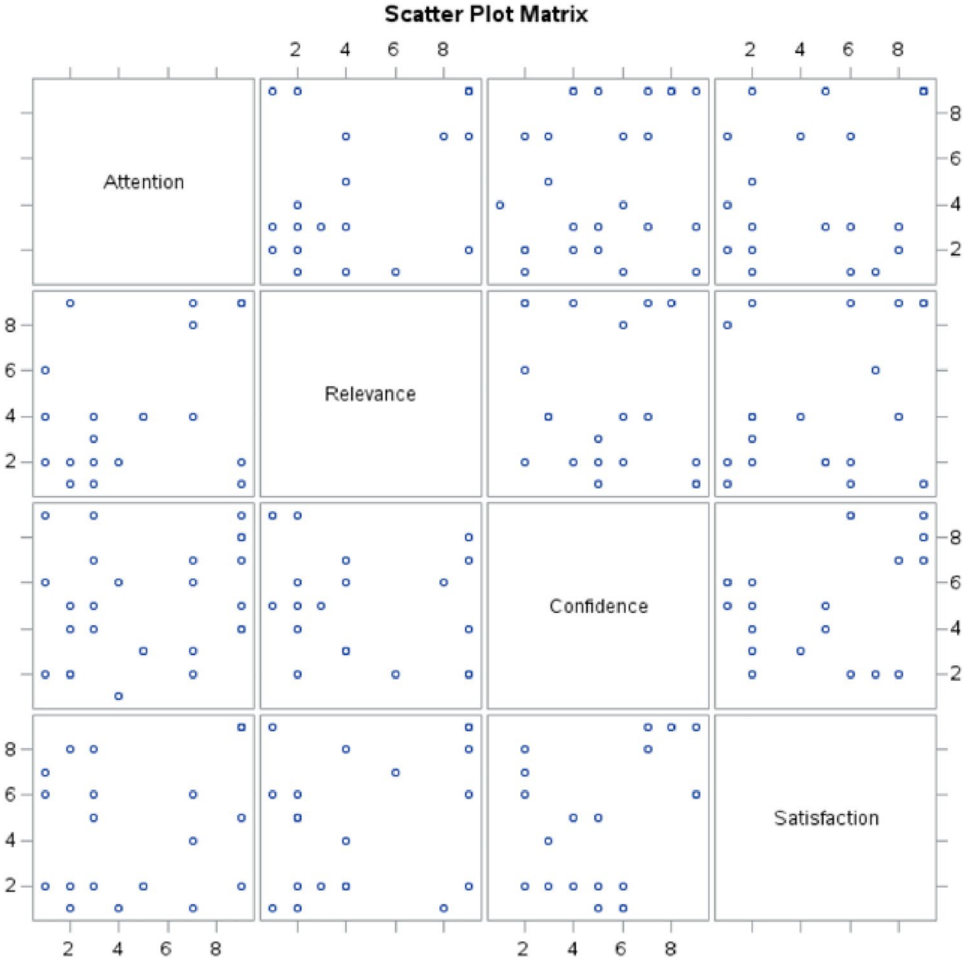

Fig. 7 The interrelation of the ARCS model

3. The resource layer, which contains data manipulation, data storage, and cloud virtualization infrastructure. The data manipulation has two algorithms: read and clean data and data integration. The read and clean data algorithm are responsible for reading the JSON files and outputting the lists of social media attributes. The data integration algorithm is responsible for integrating the data by taking the lists of social media attributes and outputting the lists of integrated data sources.
4. Finally, the visualization layer includes data summarization and a dashboard. We proposed algorithms to integrate big data from distributed data sources. The experiment results showed that the execution rate of the data retrieval algorithm for YouTube was lower than that of Facebook and Twitter. For the read and clean data algorithm, the execution rates for YouTube and Twitter were higher than that of Facebook. Finally, the data integration showed that the execution rate for Facebook was lower than that for Twitter and YouTube.

A proof-of-concept implementation of the big data integration framework used Kibana, and it had a web interface for suitable remote access. Hence, the user can perform many queries from multiple data sources as well as visualize the data integration in an appropriate format. The framework was deployed on the VMware cloud platform running on an Ubuntu operating system. We used Elasticsearch as the big data distributed and reliable storage. We developed a transformation adapter to support the transformation of the bulk data on the back-end resources. Apache Kafka was used as a back-end resource to integrate the big data in our prototype framework. The functionality of the big data integration framework was evaluated with the help of social networking data and three algorithms based on the execution time and data sizes.

In future work, big data integration research will allow researchers to use the contribution of this study as the basis for their research. Moreover, the framework for social media big data integration on the cloud can be extended to operate with commercial distributed data sources. It is possible to add more than three social networking sites to be integrated to utilize unstructured data and perform an analysis. The framework can be expanded using Apache Nifi, specifically for Twitter, in order to automate the flow of big data between systems. The proposed framework can be expanded to have a real-data processing platform such as Apache Storm and Spark.

Furthermore, another option is to use Cassandra and MongoDB as distributed storage. Apache Flink is an open-source stream processing for distributed data streaming processing applications (Shu et al. 2013). Flink can be used to establish connectivity to file systems and data storage.

Data Availability The materials used in this study are available at https://github.com/AlShomar/AlShomar-Big-Data-Integration-Framework.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abkenar SB, Kashani MH, Mahdipour E, Jameii SM (2021) Big data analytics meets social media: a systematic review of techniques, open issues, and future directions. Telemat Inf 57:101517

Ahmed SE, Aydın D, and Yılmaz E, (2021) Linear mixed-effects model using penalized spline based on data transformation methods. In: multivariate, multilinear and mixed linear models. Springer, 2021, pp. 319–341

Ahuja SP, Mani S, Zambrano J (2012) A survey of the state of cloud computing in healthcare. Netw Commun Technol 1(2):12

Akinyemi A, Sun M, Gray AJ (2020) Data integration for offshore decommissioning waste management. Automat Constr 109:103010

Al_Rabeah MH and Lakizadeh A, (2022) Gnn-ddi: a new data integration framework for predicting drug-drug interaction events based on graph neural networks

Alqarni A (2021) A secure approach for data integration in cloud using paillier homomorphic encryption

Al-Qurishi M, Alhuzami S, AlRubaian M, Hossain MS, Alamri A, Rahman MA (2018) User profiling for big social media data using standing ovation model. Multimed Tools Appl 77(9):179–201

Arer MM, Dhulavvagol PM, Totad S, (2022) Efficient big data storage and retrieval in distributed architecture using blockchain and ipfs. In: IEEE 7th international conference for convergence in technology (I2CT). IEEE 2022:1–6

Arslan AK, Tunç Z, Çolak C (2019) An open sourced software for data transformation and an application on simulated data. In: international artificial intelligence and data processing symposium (IDAP). IEEE 2019, pp. 1–6

Bettio C, Salsi V, Orsini M, Calanchi E, Magnotta L, Gagliardelli L, Kinoshita J, Bergamaschi S, Tupler R (2021) The Italian national registry for fshd: an enhanced data integration and an analytics framework towards smart health care and precision medicine for a rare disease. Orphanet J Rare Dis 16(1):1–13

Dey P, Pandit P (2020) Relevance of data transformation techniques in weed science. J Res Weed Sci 3(1):81–89

Eftekhari A, Zulkernine F, and Martin P, (2016) Binary: a framework for big data integration for ad-hoc querying. In: 2016 IEEE international conference on big data (Big Data). IEEE, 2016, pp. 2746–2753

Fillinger S, de la Garza L, Peltzer A, Kohlbacher O, Nahnsen S (2019) Challenges of big data integration in the life sciences. Anal Bioanal Chem 411(26):6791–6800

Fletcher RJ Jr, Hefley TJ, Robertson EP, Zuckerberg B, McCleery RA, Dorazio RM (2019) A practical guide for combining data to model species distributions. Ecology 100(6):e02710

https://github.com/AlShomar/AlShomar-Big-Data-Integration-Framework

Hasan FF, Bakar MSA (2021) Data transformation from sql to nosql mongodb based on r programming language. In: 2021 5th international symposium on multidisciplinary studies and innovative technologies (ISMSIT). IEEE 2021:399–403

Hilali I, Arfaoui N, and Ejbali R, (2022) A new approach for integrating data into big data warehouse. In: fourteenth international conference on machine vision (ICMV 2021), vol. 12084. SPIE, 2022, pp. 475–480

Jung H, Chung K (2021) Social mining-based clustering process for big-data integration. J Ambient Intell Humaniz Comput 12(1):589–600

Kalayci TE, Kalayci EG, Lechner G, Neuhuber N, Spitzer M, Westermeier E, Stocker A (2021) Triangulated investigation of trust in automated driving: challenges and solution approaches for data integration. J Ind Inf Integr 21:100186

Kancharala VS et al (2021) A graph based data integration and aggregation technique for big data. Turk J Comput Math Educ (TURCOMAT) 12(10):3842–3850

Keller JM (1983) Motivational design of instruction. Instructional design theories and models: an overview of their current status 1(1983):383–434

Kim S, Tom TH, Takeda M, Mase H (2021) A framework for transformation to nearshore wave from global wave data using machine learning techniques: validation at the port of Hitachinaka, Japan. Ocean Eng 221:108516

Kune R, Konugurthi PK, Agarwal A, Chillarige RR, Buyya R (2016) The anatomy of big data computing. Software Pract Exp 46(1):79–105

Li H, Deng J, Feng P, Pu C, Arachchige DD, Cheng Q (2021) Short-term nacelle orientation forecasting using bilinear transformation and iceemdan framework. Front Energy Res 9:780928

Manekar SA and Pradeepini G, (2017) Opportunity and challenges for migrating big data analytics in cloud. In: IOP conference series: materials science and engineering, vol. 225, no. 1. IOP Publishing, p. 012148

Nie W, Zhang Q, Ouyang Z, and Liu X, (2021) Design of big data integration platform based on hybrid hierarchy architecture. In: 2021 IEEE 15th international conference on big data science and engineering (BigDataSE). IEEE, pp. 135–140

NoAuthor A, (2020) Comparing business intelligence, business analytics and data analytics. [Online]. Available: https://www.tableau.com/en-gb/learn/articles/business-intelligence/bi-business-analytics

Paas FG, Van Merriënboer JJ, Adam JJ (1994) Measurement of cognitive load in instructional research. Percept Mot Skills 79(1):419–430

Pajooh HH, Rashid MA, Alam F, Demidenko S (2021) Iot big data provenance scheme using blockchain on hadoop ecosystem. J Big Data 8(1):1–26

Petri G, von Wangenheim CG, and Borgatto AF, (2017) A large-scale evaluation of a model for the evaluation of games for teaching software engineering. In: 2017 IEEE/ACM 39th international conference on software engineering: software engineering education and training track (ICSE-SEET). IEEE, 2017, pp. 180–189

Puth M-T, Neuhäuser M, Ruxton GD (2014) Effective use of pearson's product-moment correlation coefficient. Anim Behav 93:183–189

Rossi R and Hirama K, (2022) Characterizing big data management. arXiv preprint arXiv:2201.05929

Saenko I and Kotenko I (2022) Towards resilient and efficient big data storage: evaluating a siem repository based on hdfs. In: 2022 30th Euromicro international conference on parallel, distributed and network-based processing (PDP). IEEE, 2022, pp. 290–297

Shehab W, ElGokhy SM, Sallam E (2016) Rohdip: resource oriented heterogeneous data integration platform. Int J Adv Comput Sci Appl 7(9):104–109

Shi Z, Zhao G, and Liu J, (2020) Research on the model of command and decision system for big data. In: 2020 IEEE 3rd international conference on information systems and computer aided education (ICISCAE). IEEE, 2020, pp. 481–484

Shu P, Liu F, Jin H, Chen M, Wen F, Qu Y, Li B, (2013) etime: energy-efficient transmission between cloud and mobile devices. In: proceedings IEEE INFOCOM. IEEE 2013 pp. 195–199

VandanaKolisetty V and Rajput DS, (2021) Integration and classification approach based on probabilistic semantic association for big data. Complex Intell Syst, pp. 1–14

Viswanath G, Krishna PV (2021) Hybrid encryption framework for securing big data storage in multi-cloud environment. Evol Intel 14(2):691–698

Ye O, Guo R, Fu Y, and Deng J, (2022) A parallel top-n video big data retrieval method based on multi-features. In: 2022 7th international conference on image, vision and computing (ICIVC). IEEE, 2022, pp. 293–299

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations

Terms and Conditions