The Journal of Supercomputing
https://doi.org/10.1007/s11227-024-06413-1

# Data integration from traditional to big data: main features and comparisons of ETL approaches

**Afef Walha**[1,2] · **Faiza Ghozzi**[1,3] · **Faiez Gargouri**[1,3]

Accepted: 30 July 2024

## Abstract

Data integration combines information from different sources to provide a comprehensive view for making informed business decisions. The ETL (Extract, Transform, and Load) process is essential in data integration. In the past two decades, modeling the ETL process has become a priority for effectively managing information. This paper aims to explore ETL approaches to help researchers and organizational stakeholders overcome challenges, especially in Big Data integration. It offers a comprehensive overview of ETL methods, from traditional to Big Data, and discusses their advantages, limitations, and the primary trends in Big Data integration. The study emphasizes that many technologies have been integrated into ETL steps for data collection, storage, processing, querying, and analysis without proper modeling. Therefore, more generic and customized design modeling of the ETL steps should be carried out to ensure reusability and flexibility. The paper summarizes the exploration of ETL modeling, focusing on Big Data scalability and processing trends. It also identifies critical dilemmas, such as ensuring compatibility across multiple sources and dealing with large volumes of Big Data. Furthermore, it suggests future directions in Big Data integration by leveraging advanced artificial intelligence processing and storage systems to ensure consistency, efficiency, and data integrity.

**Keywords** Decision Support System · Big Data · ETL modeling · Data integration · Extract–Transform–Load

## 1 Introduction

Decision Support Systems (DSS) are interactive information systems that utilize advanced data analysis techniques to provide valuable insights for business decision-making. It assists management, operations, and planning by evaluating uncertainties and tradeoffs in decision-making. A common misconception when developing a DSS for business intelligence is that constructing a data warehouse

Extended author information available on the last page of the article

Published online: 16 September 2024

(DW) involves only integrating, storing, and processing data from various sources to offer decision-makers a multi-dimensional perspective. However, this notion needs to be more accurate.

The effectiveness of a DSS system depends on the Extract, Transform, and Load (ETL) process, which collects data from diverse sources and transforms it into a cohesive format that decision-makers can use to make well-informed decisions. This data integration process helps enhance business decision-making by providing relevant information that assists in making informed decisions. The ETL process is crucial, accounting for 80% of developing DW projects [1, 2]. This process involves extracting, cleaning, processing, and loading data into the DW. Designing the ETL process is complex, expensive, and time-consuming, making it the most challenging part of creating a DW. Prioritizing ETL process modeling is crucial for successful DW projects.

In recent years, the emergence of new online applications, devices (e.g., IoT and mobile devices, smartphones), and social media platforms (such as Instagram, Facebook, and Twitter) has resulted in the generation of Big Data [3]. Critical aspects of Big Data include volume, variety, velocity, and veracity [4]. This has led to significant challenges related to handling large volumes of data, its speed, and various types (unstructured, semi-structured, or structured). These challenges also encompass the need for fast data processing and ensuring data accuracy and scalability.

Numerous studies examine ETL modeling approaches. Most are tailored to handle data from the organization's traditional systems, while others address ETL approaches for Big Data integration [1]. Our research clearly focuses on the impact of evolution of data integration from transactional to Big Data systems, especially on ETL processes modeling and trends. Traditional business intelligence processes, particularly ETL, need to be reassessed to accommodate such data in adequate storage, as they are impractical for storing and processing large-scale datasets. Recent research has brought a distributed computing environment supported by frameworks (e.g., Hadoop), integrating programming models (e.g., MapReduce), and data storage systems (e.g., HDFS) to enable efficient data processing and storage. While many technical ETL solutions have been proposed in this field, they often overlook the crucial aspect of conceptual modeling in ETL processes.

This paper aims to survey ETL approaches in response to technological advancements and organizational modeling needs arising from data evolution. It seeks to explore the benefits and weaknesses of each approach, typical use cases, and the challenges and trends encountered during the data transition. Our search method includes formulating research questions, identifying keywords, developing search strings, and reviewing papers based on inclusion/exclusion criteria to select the most relevant references. Our study leads to a comprehensive literature review of ETL modeling approaches from the early 2000s until the beginning of 2024. The goal is to assist researchers in data integration and various organizational stakeholders, such as data warehouse and ETL designers, data engineers, project managers, and business analysts, in choosing the most suitable approach for their needs.

In this survey, various advantages, issues, and challenges in this field are discussed for ETL approaches. This paper summarizes the exploration of ETL modeling, focusing on reusability, user requirements, ETL steps, data scalability, and processing. It also identifies critical dilemmas and trends, such as maintaining data scalability and ensuring compatibility across multiple sources and large volumes of Big Data. It suggests future directions by proposing useful advanced technologies like natural language processing (NLP), artificial intelligence (AI), and large language models (LLMs) to improve automation, speed, efficiency, and quality in data integration processes.

The main contributions conducted in this paper may be summarized as the following points:

1. Providing a comprehensive overview and classification of current literature on ETL modeling approaches, considering the transition from traditional to Big Data.
2. Analyzing the evolution of ETL approaches in response to technological advancements and organizational needs caused by Big Data integration.
3. Evaluating the effectiveness of different ETL approaches according to the main modeling criteria (e.g., reusability, flexibility, meeting business requirements, scalability) in addressing data integration challenges.
4. Discussion and recommendation for future modeling directions and practical solutions in the field of Big Data integration.

This paper is structured in the following manner: Section 2 overviews research method. Section 3 overviews our proposed classification of ETL approaches. Section 4 outlines our criteria for comparing and evaluating the literature review. Section 5 summarizes pertinent studies on modeling ETL processes for traditional data based on specific formalism or standard notations and modeling. We also evaluate these contributions. Section 6 comprehensively compares various Big Data technologies used in ETL processes. We also present some ETL modeling solutions in the context of big data. Section 7 discusses the literature review and highlights current trends and future directions. Section 8 concludes the study and provides suggestions for future research.

## 2 Research method

The process for selecting papers according to [5] involves the following steps:

- Formulating multiple research questions based on the research area.
- Using the research questions to determine keywords for the search.
- Creating search strings based on the keywords.
- Checking the final papers based on inclusion and exclusion criteria.

### 2.1 Research questions

In this research, we analyze and examine data integration and ETL process approaches. We intend to address the research questions listed below:

(Q1) What are the main ETL approaches for integrating transactional data?
(Q2) What are the pros and cons of ETL modeling solutions regarding reusability, user requirements, modeling aspects, and the design and implementation of ETL steps?
(Q3) What are the primary deficiencies of traditional ETL approaches in handling various data sources generating vast volumes of data? What challenges are commonly encountered during transitioning from traditional ETL to Big Data, and how can they be effectively mitigated in a distributed computing environment?
(Q4) What are the current and future trends in Big Data integration?

### 2.2 Search strategy

We used search queries to find literature reviews. The search strings were created after formulating the study questions, and we selected keywords based on the research questions. The keywords detected were "Decision system", "Extract-transform-load", "Data integration", "ETL process", "Big Data", and "Modeling," etc. We initially obtained survey findings using the search query (e.g., "ETL OR data integration" AND "modeling"). The search language was modified for individual needs. The databases for references were Google Scholar, Springer, IEEE Xplore, Science Direct, and ACM Portal. The search was conducted from 2001 to the beginning of 2024. Figure 1 displays the word cloud generated by analyzing the references and abstracts. It features the top 100 most frequently used terms, with word size corresponding to the frequency of usage.

### 2.3 Search selection

Upon receiving the database findings, it is crucial to thoroughly examine each document to verify its relevance to our survey setting. The studies were identified by applying the following inclusion and exclusion criteria. Figure 2 displays the outcomes obtained at each step. Step 1 involves applying the search query to all available sources and collecting the results. Step 2 involves applying inclusion/

**Fig. 1** References-abstract word cloud

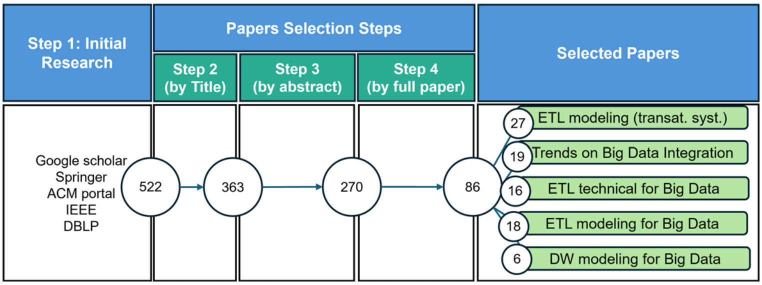


**Fig. 2** Papers selection

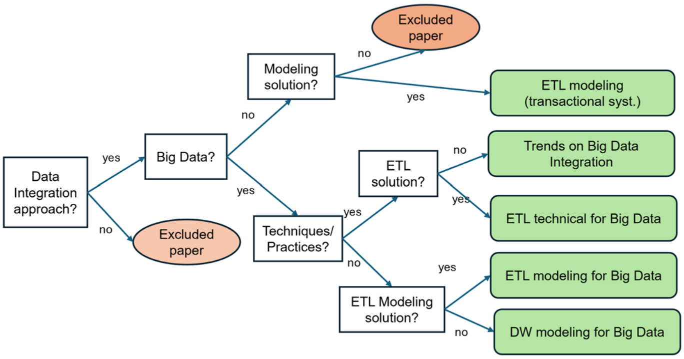


**Fig. 3** Decision tree- final papers selection (Step 4)

exclusion criteria to the article title. Step 3 involves applying inclusion/exclusion criteria to the abstract and introduction of the paper. Step 4 involves applying inclusion and exclusion criteria to the entire paper. Figure 3 illustrates the scenario followed to select the final papers.

The inclusion criteria are as follows:

- Studies are addressing ETL modeling approaches (specific or standard-based) for all type of data sources: structured, semi-structured or unstructured data.
- Studies are addressing technical or ETL modeling solutions for Big Data integration.
- Studies use predetermined performance measures to assess their performance approach.
- Studies are published in journals, or book or presented at conferences in the last decade

The exclusion criteria are as follows:

- Studies that do not focus on the data integration process (analysis or querying).

- Studies that do not present a modeling solution for transactional systems.
- Studies that are not relevant to the research questions.
- Studies conducted before 2000.

## 3 Classification of data integration approaches

In this literature review, we evaluate the work that has been conducted in the data integration process and emphasize the most significant contributions. We have summarized our research in this area from the early 2000s to the beginning of 2024 in Fig. 4. To ensure inclusivity and cover a broad spectrum of research in the literature, we have categorized these research works into four primary categories based on the modeling formalism, data type, and technologies they utilize.

(1) Specific ETL process modeling approaches for transactional systems.
(2) Standard-based ETL modeling approaches for transactional data.
(3) ETL technical approaches for Big Data.
(4) ETL process modeling approaches for Big Data.

Approaches in (1) involve defining specific notations for transactional systems (especially handling structured data), while those in (2) utilize modeling standards or notations (e.g., UML, BPMN) for the same data system. Approaches in category (3) focus on exploring specialized architectures and technologies for integrating Big Data, while approaches in (4) offer modeling solutions for the complex tasks and processes involved in extracting big data, formatting it, and transforming it into the

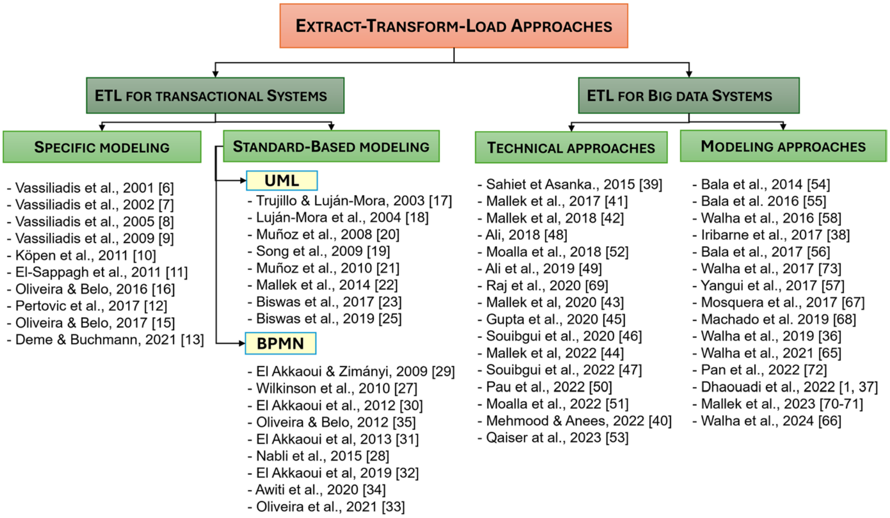

Extract-Transform-Load Approaches
ETL for transactional Systems
ETL for Big data Systems
Specific modeling
Standard-Based modeling
Technical approaches
Modeling approaches
- Vassiliadis et al., 2001 [6]
- Vassiliadis et al., 2002 [7]
- Vassiliadis et al., 2005 [8]
- Vassiliadis et al., 2009 [9]
- Köpen et al., 2011 [10]
- El-Sappagh et al., 2011 [11]
- Oliveira & Belo, 2016 [16]
- Pertovic et al., 2017 [12]
- Oliveira & Belo, 2017 [15]
- Deme & Buchmann, 2021 [13]
UML
- Trujillo & Luján-Mora, 2003 [17]
- Luján-Mora et al., 2004 [18]
- Muñoz et al., 2008 [20]
- Song et al., 2009 [19]
- Muñoz et al., 2010 [21]
- Mallek et al., 2014 [22]
- Biswas et al., 2017 [23]
- Biswas et al., 2019 [25]
BPMN
- El Akkaoui & Zimányi, 2009 [29]
- Wilkinson et al., 2010 [27]
- El Akkaoui et al., 2012 [30]
- Oliveira & Belo, 2012 [35]
- El Akkaoui et al, 2013 [31]
- Nabli et al., 2015 [28]
- El Akkaoui et al, 2019 [32]
- Awiti et al., 2020 [34]
- Oliveira et al., 2021 [33]
- Sahiet et Asanka., 2015 [39]
- Mallek et al., 2017 [41]
- Mallek et al, 2018 [42]
- Ali, 2018 [48]
- Moalla et al., 2018 [52]
- Ali et al., 2019 [49]
- Raj et al., 2020 [69]
- Mallek et al, 2020 [43]
- Gupta et al., 2020 [45]
- Souibgui et al., 2020 [46]
- Mallek et al, 2022 [44]
- Souibgui et al., 2022 [47]
- Pau et al., 2022 [50]
- Moalla et al., 2022 [51]
- Mehmood & Anees, 2022 [40]
- Qaiser at al., 2023 [53]
- Bala et al., 2014 [54]
- Bala et al. 2016 [55]
- Walha et al., 2016 [58]
- Iribarne et al., 2017 [38]
- Bala et al., 2017 [56]
- Walha et al., 2017 [73]
- Yangui et al., 2017 [57]
- Mosquera et al., 2017 [67]
- Machado et al. 2019 [68]
- Walha et al., 2019 [36]
- Walha et al., 2021 [65]
- Pan et al., 2022 [72]
- Dhaouadi et al., 2022 [1, 37]
- Mallek et al., 2023 [70-71]
- Walha et al., 2024 [66]

**Fig. 4** Classification of ETL approaches

target decision system. As illustrated in Fig. 5, ETL for Big Data is relatively new, having begun in 2014, compared to transactional systems.

These methods differ in their capacity to handle data and complexity based on the technologies and paradigms adopted. However, their primary aim is to make data extraction, transformation, and loading as fast, efficient, and straightforward as possible. Each method has pros and cons, and choosing one depends on several factors, such as the data type and the organization's requirements. The objective is to manage data in a manner that is both effective and secure enough for use in decision-making systems.

### 3.1 ETL for transactional systems

Transactional systems or OLTP (Online Transaction Processing) systems are designed to simultaneously handle numerous daily small transactions, such as sales, orders, and financial transactions. They maintain data integrity and consistency using ACID (Atomicity, Consistency, Isolation, Durability) properties. They use relational databases for structured data management and enable real-time processing, resulting in faster data processing and query response times.

Transactional databases are a vital type of database management system that is highly optimized to quickly read and write individual rows of data while ensuring data integrity. Traditional data integration approaches focus on the ETL methods for transactional systems that efficiently manage and integrate structured data for business operations and analytics. The main ETL steps are described as follows:

- Extraction: This involves collecting data (e.g., Sales transactions) from company data sources, usually stored in multiple relational databases and operational systems such online order databases.
- Transformation: This step involves data cleaning to ensure data quality, such as identifying and correcting errors, inconsistencies, and missing values.

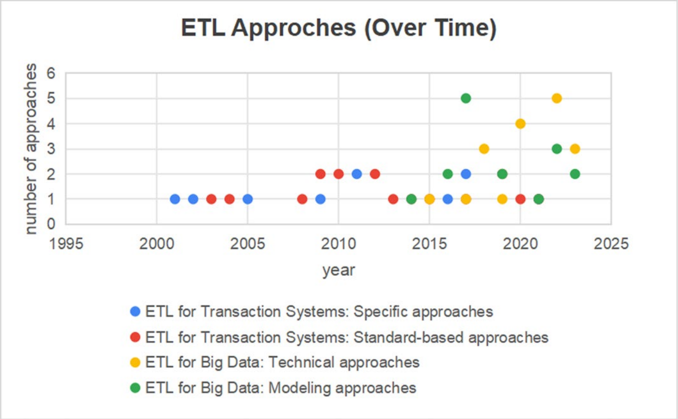


**Fig. 5** Classification of ETL approaches (over time)

Combining data from various sources to create a unified view often involves deduplication, normalization, and integration of disparate data formats. Business rules may be applied to transform the data into a meaningful format for analysis and reporting.

- Loading: The transformed data is placed into data warehouses (which serve as a centralized data repository), data marts, or other analytical databases for querying and reporting.

OLTP is limited in its ability to handle large-scale data analysis and diverse data types because it is designed for real-time transaction processing and structured data. Big Data systems, on the other hand, can efficiently store, process, and analyze large amounts of diverse data using distributed architectures and powerful frameworks like Hadoop and Spark for both batch and real-time processing [1].

### 3.2 ETL for Big Data systems

Big Data encompasses a wide range of information, including media, imaging, audio, sensor data, text data, and various other data types. Such data is characterized by four key features (known as Big Data 4 Vs): Volume, velocity, variety, and veracity [4]. Volume refers to the amount of data collected; velocity is the speed at which it is created and collected; variety refers to the scope of data points covered, and veracity is the data's accuracy and quality.

Big Data is collected from heterogeneous and various sources, such as user's comments on social media, apps, reviews, and product purchases, that can provide insights into customer needs. So, we are talking about more than just structured data stored in databases and spreadsheets or semi-structured data, which has some structure but does not conform to a specific data model. In the modern world of big data, unstructured data, a vast collection of files that lack a predefined data model or format, is the most prevalent. The content can originate from either humans or machines, and it can be in the form of text or other formats.

ETL for Big Data is a complex process of extracting data, especially from unstructured sources, transforming it, and loading it into a target decision-making database or data warehouse. Thus, it requires specialized tools and technologies to ensure data is appropriately formatted, cleansed, and validated before loading it into the target system. This process is crucial to the success of big data integration projects, as it enables businesses to make informed decisions based on accurate and reliable data.

### 3.3 Integrating Big Data into traditional ETL methods: defects and challenges

Traditional ETL methods may face several deficiencies when dealing Big Data systems, primarily due to their design and capabilities tailored for smaller datasets and structured data. Some critical defects include:

- *Scalability issues* Traditional databases and processing systems may struggle to scale horizontally in Big Data environments, impacting efficiency in managing increasing data loads.
- *Performance constraints* Due to rising data volumes, traditional methods can lead to slower response times and reduced system performance.
- *Data variety challenges* Integrating and processing structured, semi-structured, and unstructured data forms is challenging with traditional methods.
- *Cost and complexity* Scaling traditional systems for Big Data can be expensive and complicated, involving hardware upgrades, software licensing, and specialized management skills.
- *Limited analytical capabilities* Traditional methods may lack advanced capabilities for extracting insights from Big Data, such as machine learning and predictive analytics.

The highlighted drawbacks accentuate the necessity of employing specialized modeling approaches, tools, architecture, and technologies customized for Big Data integration. This encompasses distributed computing frameworks (such as Hadoop and Spark), large-scale data storage and management systems (like NoSQL databases, cloud-based storage, data lakes), and an appropriate modeling environment such as standards and notations, meta-models, and models [3].

## 4 ETL approaches: comparing criteria and features

We outline several essential criteria for conducting a comparative study of research papers focusing on ETL process modeling and Big Data integration. Most of these criteria are familiar in the context of ETL modeling, while the current research has inspired others in the field of Big Data integration.

We chose the following criteria for our literature review of the existing ETL modeling approaches:

- *Modeling language* refers to the language or notation used for modeling the ETL phases. It involves specific notations or conventional standards (such as UML or BPMN).
- *Meta-modeling* checks to see if the proposed models of ETL activities, operations, and processes are validated against the constraints defined in meta-models. This point makes the ETL process more accurate and reliable.
- *Modeling level* represents the level of abstraction (conceptual (C), logical (L), and physical (P)) used to model the ETL process.
- *Visualization flow* conveys which parts of the modeling aspect (e.g., control flow, data flow, expression, template, and data sources) should be considered when designing an ETL process.
- *Data sources* reveal level of structure of the data as well as their origin of data (human-generated content, external data, social media).

- *Target* is the system for storing the output data of the ETL process. A target system can take many forms, including data warehouses, data marts, reports, NoSQL databases, or other decision-making formats.
- *ETL models* enumerate samples of models suggested to the basic ETL activities, operations, or tasks.
- *ETL implementation* examines whether components/tasks have been implemented to put into action ETL-designed models.

To simplify the comparison of ETL approaches for BigData, we have formulated additional criteria. These criteria are as follows:

- *Big Data Vs* refers to the different aspects of Big Data (Vs: velocity, veracity, volume, variety) handled by the ETL approach.
- *Big Data technology* refers to the specific technology (such as MapReduce and Hadoop) that the ETL approach employs.
- *Architecture* examines whether the approach is designed to function in a distributed or parallel environment for Big Data.

We have reviewed the most important research papers focused on the ETL process or activities, as well as the ideas, modeling formalisms, concepts, tools, architecture, and technologies they used. Following this, we compared them using specific criteria and previously detailed features. Each category of approaches studied is presented in a comparison table with the comparison criteria in columns and the different contributions in rows. Tables 1, 2, 3, 4 and 5 compare ETL approaches.

## 5 ETL for transactional data

The primary objective of traditional ETL methods is to design the extraction, transformation, and loading of classical data, focusing on companies' transactions. These methods can be classified into two categories: specific and standard-based.

### 5.1 Specific ETL modeling approaches

Regarding ETL modeling, specific design involves creating ETL activities and processes without relying on established modeling standards or languages. This approach allows developers to introduce new concepts and notations that can effectively represent the intricate flow of data transformations in data integration.

#### 5.1.1 Summary of specific ETL modeling approaches

Vassiliadis et al. introduced ARCTOS as a tool for ETL processes, providing primitives for data cleaning, scheduling, and transformations [6]. In [7], they provided formal foundations for their graphical representation of ETL activities.

In [8], they defined a meta-model for logical entities incorporated into ARCTOS II, a design tool for traditional DW flows. Finally, they identified generic properties of ETL activities using a black-box approach and provided a taxonomy based on input–output relationships [9]. In the same context, Köppen et al. [10] proposed pattern formalization to relate ETL process design to software engineering. They introduced patterns for complex tasks like aggregation, data changes, and duplication, saving time and making the process user-friendly. This approach offers straightforward solutions for recurring ETL tasks but requires a user-understandable problem, context, and solution.

Unlike previously described approaches focusing on structured data sources, El-Sappagh et al. [11] proposed the entity mapping diagram (EMD), a conceptual model for designing the ETL process for unstructured data. EMD is organized into data source extraction, mapping functions, and DW schema (where data facts and dimensions are loaded). EMD offers user-defined functions (UDF) that can be easily integrated with the development environment. Petrović et al. [12] considered process flows differently from the ETL data flow design aspect that [6–11] employed. They also suggested three additional domain-specific languages (DSLs) that relate to different aspects of ETL modeling: ETL-E (logical and arithmetic expressions), ETL-T (transformation patterns), and ETL-D (modeling source and target data). Each DSL's corresponding concepts and rules are defined in a separate meta-model with clear and precise semantics, simplifying the process's design complexity. Recently, Deme and Buchmann [13] discussed the limits of traditional DSLs and focused on technology-specific conceptual modeling languages to achieve interoperability with a technological environment, using interfaces and abstractions that must be orchestrated at runtime. Fast prototyping platforms based on meta-modeling allow for agile deployment and offer benefits even when reusability outside the application context is limited.

In reference [14], software design patterns are pre-established tasks grouped by context, allowing for reconfiguration and preventing the need to rewrite repetitive tasks and activities. The authors proposed a pattern-oriented approach to support various phases of an ETL lifecycle. The ontology hierarchy enables the expression of ETL patterns using classes, data properties, and object properties. Additionally, it provides the basic structure to support the development of a specific language for pattern instantiation. Authors in [15] introduced a new ontology graph to summarize the fundamental concepts behind imprint structure and configuration. However, further development is needed to clarify the DS and DW schemes and the mapping between the two compartments.

### 5.1.2 Comparison of specific ETL approaches

As explained in Sect. 5.1.1, ETL-specific methods suggest notations, concepts, and languages for the conceptual, logical, and physical levels. These methods also introduce new models for standard ETL tasks like filter, union, join, and convert. Combining these models makes it more efficient to design intricate ETL processes.

Most of the approaches (compared in Table 1) focus on modeling ETL data flow. These studies showcase ETL activities and their input and output data, such as [6–9,

**Table 1** Comparison of specific ETL modeling approaches

| Approaches | Modeling Language | Meta-model | Modeling level | | | Visualization Flow | Data Sources | ETL models | ETL implementation |
|---|---|---|---|---|---|---|---|---|---|
| | | | C | L | P | | | | |
| [6] | ETL concepts | x | x | x | x | Data flow | Struct. | Delete, Insert to table, Null-exist, Format violate | ARKTOS (SQL queries) |
| [7] | Graphic notation | x | x | | | Data flow | Struct. | Surrogate keys, Check null_values | - |
| [8] | Specific concepts | x | | x | x | Data flow | Struct. | NotNull, Domain Mismatch | ARKTOS II |
| [9] | Specific notation (taxonomy) | | | x | x | Data flow | Struct. | Compose, Couple, Swapp | yes |
| [10] | ETL patterns | | x | | | Data flow | Struct. | Aggregate, Duplicate, Convert, History | - |
| [11] | specific notation | x | x | | x | Data flow | Struct./ Un-struct. | Filter, Union, Join | EMD builder |
| [12] | DSLs | x | x | | | Process, Operations, Data, Template, Expression | Struct. | Convert | yes |
| [13] | technology-specific language | x | x | | | Data/ Process flows | Struct./ Semi-struct. | Filter, Set fields, Generate missing pair | code generator |
| [14; 15] | ontology | x | x | x | x | Process flow | Struct./ Un-struct. | Aggregator, Conciliation | pattern (OWL) |

11]. However, they do not consider the order of their execution (process flow). This issue is addressed in the study [13], which controls how ETL activities are defined in a complete process. Additionally, authors in [12] have developed separate visualization aspects (process, operation, data, template, and expression) for the ETL process.

ETL approaches differentiate between conceptual and logical design levels, with conceptual design (C) aiming for an implementation-independent schema and logical design (L) creating a logical schema. In addition, other methods (e.g., [11]) support physical design (P) for implementation issues. Our study shows that the conceptual level (C) is commonly considered. Meta-models structure the concepts and notations of specific languages in most approaches, providing generic and customized ETL modeling solutions. Each meta-model showcases one or multiple visualization aspects of the ETL process.

Our research emphasizes the significance of framing ETL tasks properly. Deme and Buchmann [13] enables automated code generation, which shortens development time and increases productivity. Furthermore, it ensures that the executed activities align closely with their respective application domains.

Specific ETL design approaches offer flexible, generic, customized models for ETL processes and activities, considering several modeling aspects. However, they do not utilize standard notations. To this end, ETL designers must be familiar with these notations and concepts to define or adapt the ETL model. To address this issue, approaches (discussed in Sect. 5.2) suggest ETL design based on conventional modeling standards.

### 5.2 Summary of standard-based ETL modeling approaches

Designers can simplify the ETL process using familiar modeling standards such as UML or BPMN. Since they do not have to learn new concepts specific to ETL process modeling, this method helps them save time and concentrate on ensuring the accuracy of ETL processes.

#### 5.2.1 Summary of ETL modeling approaches based on UML

Unified modeling language (UML) is a standardized language widely used in software engineering to illustrate and document software systems [16]. ETL developers often utilize UML to represent the different components and their relationships, which aids in comprehending and communicating complex ETL processes.

In [17], authors were among the first to use UML class packages to model the ETL process. They also created a set of UML stereotypes to represent the most common ETL tasks, like combining different data sources, changing attributes between source and target, and making surrogate keys. Similarly, Lujan-Mora et al. [18] created a data mapping diagram that connects data sources to the DW (data flows). However, the communication and message exchange structure between processes is absent, and there are no mechanisms for representing specific conditions, such as control flow sequences and temporal constraints.

Song et al. [19] proposed a method for modeling the ETL process at a conceptual level using the extended UML profile mechanism. They created two ETL meta-models, namely the data meta-model and the ETL operation meta-model, by defining UML profiles specific to typical operations like wrapper, merge, and join. This approach reduces an ETL tool's cost and development cycle. However, its static aspect must reflect the concepts of chaining ETL operations and exchanging data between them. To improve design flexibility, some researchers have used the UML activity chart. Authors in [20] created a conceptual modeling framework to describe ETL behavioral needs. It was similar to the approach [17], but there were significant changes in time constraints, dynamic aspects, and control flows. The framework consists of two layers: a high-level layer that specifies elements and a low-level layer containing ETL workflows. Experimentation showed that the activity diagram is valuable for illustrating ETL's dynamic elements [21]. Despite the benefits of the suggested activities, they are limited to internal data source integration.

To incorporate web data into decision-making support, Mallek et al. [22] extended the approach [20] proposed ETL process conceptualization method. ETL models were proposed for the main ETL steps, such as collecting websites and clickstreams and their transformation (cleaning, filtering, and mapping) into the DW. This approach was the first to consider weblogs' importance in ameliorating companies' decisions. Nevertheless, it does not propose a model of the whole ETL process. Activity models were designed using an ETL-Web meta-model, which extends the UML activity meta-model with concepts specific to the web data context (such as "log file" and "website"). Each model has been implemented as an ETL component in Talend Open Studio (TOS). Biswas et al. [23] proposed an ETL method that uses SysML (extension of UML [24]), to investigate requirements and activity diagrams. Afterward, they extended this work in [25] by demonstrating how to automate the validation of SysML models. They used a case study of an e-commerce website to illustrate the method. The simulation results showed that the proposed method for validating the ETL model was effective.

### 5.2.2 Summary of ETL modeling approaches based on BPMN

BPMN (Business Process Modeling and Notation) is a valuable standard for modeling business processes and web services [26]. It offers a straightforward notation that experts such as analysts, developers, and designers can understand. Considering ETL as a particular type of business process, many researchers have found that the BPMN notation is beneficial in ETL process modeling.

Wilkinson et al. [27] proposed a multi-layer approach that uses three layers: a business requirements layer, a conceptual layer, and a logical layer. These layers are based on the BPMN standard and aim to improve the quality of ETL process modeling. Nabli et al. also used BPMN in [28] to introduce a two-phase ETL design method for DW development. The first phase matches the source target and identifies transformation operations to create a correspondence table. The second

**Table 2** Comparison of standard-based ETL modeling approaches

| Approaches | Modeling Language | Meta-model | Modeling level | | | Visualization Flow | Data Sources | ETL models | ETL implementation |
|---|---|---|---|---|---|---|---|---|---|
| | | | C | L | P | | | | |
| [17] | UML stereotype | | x | | | Data flow | Struct. | Surrogate, Merge, Insert multi-table | yes |
| [18] | UML packages | | x | | | Data flow | Struct. | aggregate, filter, divide | - |
| [19] | UML profile | Data/ Operation | x | | | Data flow | Struct. | Wrapper, Filter, Clean | - |
| [20] | UML activity diagram | | x | | | Data/Control flows | Struct. | Incorrect, Loader, Conversion, Merge | - |
| [21] | UML activity diagram | | x | | | Data/Control flows | Struct. | Surrogate Sales, Log, Product loader | - |
| [22] | UML activity diagram | ETL-Web | x | | | Data/ Control flows | Semi-struct. (blogs, web sites) | Clean logs, Web transformation, Business-web mapping | ETL-web (TOS) |
| [23] | SYSML | | x | | | Data/ Process flows | Semi-struct. | ETL tasks (e-commerce sales) | - |
| [25] | SYSML | | x | | | Data/ Process flows | Semi-struct. | ETL tasks (e-commerce sales) | Simulation of SYSML model |
| [27] | BPMN | | x | x | x | Process | Struct. | DailyRevenue | Physical implement. |
| [28] | BPMN | | x | | | Process flow | Struct. | Discriminate, Decompose, Explosion | yes (TOS) |
| [29] | BPMN | yes | | | x | Data/ Control flows | Struct., Semi-struct. | Load geography and time dim., Load sales fact | yes (BPEL) |
| [30] | BPMN | BPMN4ETL | x | | | Data flow | Struct. | Load category fact | yes (OMB) |
| [31] | BPMN | | x | | | Data/ Control flows | Struct., Semi-struct. | DimGeography load | code generation (OMB) |
| [32] | BPMN | SPK pattern | x | | | Control flow | Struct. | Surrogate Key pipeline | BPEL execution |
| [33] | BPMN | | x | | | Process flow | Struct. | Extract and Load customer | yes (pentaho, TOS) |
| [34] | BPMN | | | x | x | Data/ Control flows | Struct., Semi-struct. | SDC | yes (Pentaho, TOS) |

phase implements the ETL process, minimizing complexity and benefiting from the knowledge stored in the table.

In [29], El Akkaoui et al. used a subset of graphic constructors of BPMN symbols to design ETLs. This approach transformed BPMN conceptual notation and the BPEL (Business Process Execution Language) to ensure ETL execution. In [30], they proposed an ETL modeling framework based on model-driven architecture (MDA). They viewed an ETL process as combining two perspectives: control and data processes. ETL objects were organized into a data process meta-model (BPMN4ETL). Authors in [31] proposed a model-driven automatic code generation and language maintenance framework using Oracle Meta Base (OMB). They demonstrated validation using a practical example. Then, they completed this approach in [32] and provided a set of internal and external measures of the design quality of the ETL process. ETL components are defined on Microsoft SQL Server Integration Services (SSIS).

In [33], Oliveira and Belo expanded on the work of [29] and [30] by developing conceptual models for adaptable data that can be applied to any ETL standard process. They explored the use of BPMN in designing and testing ETL processes and created a meta-model that serves as the template for the Surrogate Key Pipeline (SPK) ETL process. Recently, authors in [33] proposed a set of guidelines for ETL conceptual modeling that use BPMN notation more consistently. This approach breaks down ETL conceptual modeling into different layers (process, pattern, and task), each representing a different level of detail in the process and providing specific tools for communication within the ETL development team during different phases. This proposal contributes to a more efficient development process, as models can be incrementally improved to meet system requirements.

Awiti et al. [34] proposed a method for developing an ETL process for updating slowly changing dimensions (SDC) using relational algebra. This method is based on the BPMN4ETL conceptual model, first introduced in [30] and then transformed into extended relational algebra (RA) with SDC operations. The study also presents a translation mechanism from BPMN to the RA specification. When SDC operations were tested on Pentaho data integration and Talend Open Studio, they showed that relational algebra worked better with ETL tools than ETL4BPMN when making the ETL flow.

### 5.2.3 Comparison of standard ETL approaches

Standard-based approaches have contributed to the design of the ETL process using conventional modeling languages and notations, where concepts are understandable and well-structured. Such approaches are compared in Table 2 based on criteria (detailed in Sect. 4).

UML and BPMN are the most widely used standards for ETL process modeling. UML-based approaches offer a breakdown of complex ETL processes into typical activities (e.g., aggregation, fusion, and filtering). Adapting or extending UML meta-models provides valid generic models designed for these activities. Even though they are consistent, most UML-based approaches (e.g., [18, 19, 22]) have only examined how data moves during ETL tasks (data flow), not how important the control flow is. Based on the idea that an ETL process is a particular instance of a business process, several approaches used BPMN notations, offering rich solutions to cover all stages of the ETL process. Most of these approaches emphasized separating data and process flows when modeling the ETL process [29–32, 34].

The comparison of standard modeling approaches (established in Table 2) shows that several approaches focus solely on ETL activities data flows, but this oversimplifies the complexity of the ETL process. Separating modeling aspects (such as control flows, data flows, and data modeling) is essential to create a more dynamic and flexible approach. Unfortunately, some models (e.g., [17, 18]) lack mechanisms for specific conditions like control flow sequences and time constraints, leading to static and rigid modeling. As exemplified by [12, 21, 22, 32–34], a better approach is to adhere to the sequence of ETL activities to incorporate dynamic aspects in ETL process design.

### 5.3 Standard versus specific ETL modeling approaches

We distinguish two main types of ETL modeling approaches for traditional data integration. The ETL modeling approaches (explained in Sect. 5.1) propose specific concepts and notations for designing the ETL process, providing formal and conceptual models for complex ETL flows. In contrast, Standard-based approaches (presented in Sect. 5.2) adapt or extend existing standards or modeling languages, such as UML, SysML, and BPMN.

The study reveals that using specific concepts and notations in ETL process modeling benefits clarity, efficiency, and collaboration. These notations ensure that all stakeholders understand the unique aspects of ETL accurately. However, these approaches define non-standard notations, so ETL designers must become familiar with these concepts. This familiarity is a limitation of specific approaches to effectively define or adapt the ETL model. Standard-based approaches, such as UML and BPMN, offer universality, interoperability, flexibility, and long-term support. They are practical choices for data integration projects, documentation, maintenance, and governance of complex ETL systems.

The ETL modeling approaches we previously studied offer practical conceptual modeling solutions for transactional data, regardless of the employed specific or standard notations. They facilitate the design and understanding and reusability of complex ETL processes and tasks and enable automatic code generation. A critical aspect of the modeling is the separation of flow visualization to simplify the modeling of complex ETL processes. It is evident in Tables 1 and 2 that most approaches (e.g., [13, 20–24, 29–31, 34]) emphasize the separation of data flows and control flows. Moreover, other approaches, such as [12], considered additional flows like expressions and templates.

Various studies have proposed different methods to improve the reusability and adaptability of complex ETL (Extract, Transform, and Load) processes. However, there is a clear gap in research regarding meta-modeling that would empower organizations to create, implement, and manage efficient and scalable ETL processes. The utilization of meta-models to structure ETL models based on standards such as [10, 17, 18, 20, 21, 24, 34] could enhance the solution, ultimately leading to more effective data integration and improved decision-making.

ETL modeling approaches compared in Tables 1 and 2 both concentrate on the conceptual level (C); however, there is a lack of research on the logical (L) and physical (P) levels to have a complete ETL modeling solution. On the other hand, our analysis of ETL modeling solutions revealed that ETL modeling efforts are predominantly data-driven. This point implies that only data sources are considered during extraction, transformation, and loading without focusing on the user's requirements. Little works (e.g., [11, 28]) considered the designer's needs when defining their transformation operations.

Ultimately, standard and specific ETL modeling approaches offer useful templates for ETL designers to define complex ETL processes. These approaches offer valid models for transforming business data sources into a data warehouse. They accommodate traditional decision support systems with limited and structured data sources stored in a relational database. However, with the increasing prevalence of Big Data stored in NoSQL databases, it is necessary to consider whether traditional ETL processes, operations, models, frameworks, and techniques are still effective in aiding companies making better business decisions. Several tasks within the ETL processes need to be redefined to align with Big Data environment, emphasizing new issues such as data volume and heterogeneity [36, 37].

# 6 ETL for Big Data integration

Conventional decision-making systems are not capable of handling "Big Data" through the ETL process modeling methods that were previously effective. New techniques are required to manage the enormous amount of unstructured or semi-structured data. Researchers have developed new ETL systems to assist traditional operators in designing schema-less databases and resolving the problem.

## 6.1 Technical ETL approaches for Big Data

NoSQL systems are used for Big Data and fall into four categories: key-value, record/column-oriented, document-oriented, and graph-oriented [38]. New techniques and strategies have been developed to integrate NoSQL databases effectively into data warehouses.

### 6.1.1 Summary of technical Big ETL approaches

Over the last decade, many researchers have highlighted the limited support for unstructured and semi-structured data. Authors in [39] have studied ETL frameworks for migrating relational systems to NoSQL, identifying necessary join operations for real-time data warehousing of unstructured data streams. In [40], Mehmood and Anees developed an ETL architecture for managing Big Data in real-time and distributed environments. They proposed a stream-disk join operation to simultaneously process unstructured stream data and disk data from dispersed stores. This

**Table 3** Comparison of ETL technical approaches for Big Data

| Approaches | Data Sources | Target | ETL operation/ tasks | ETL implementation | Big Data Vs | Big Data technology | Architecture |
|---|---|---|---|---|---|---|---|
| [42, 43, 44] | Semi-struct. (Tweets) | DW (column-oriented DB) | Select, Project, Join, Convert | BigDimETL (Hive, Hbase), TOS | Volume | MapReduce | Distributed/ Parallel |
| [45] | Struct. (Web pages) | DW (column-oriented DB) | Filter, Tokenize Field, Sort DB, Match records | yes | Volume, Variety | MapReduce | Distributed |
| [46, 47] | Semi-Struct. | DW (JSON documents) | Join multiple documents | yes | Variety | - | - |
| [49] | Semi-Struct., Un-struct. | DW | Sentiment analysis generator | Yes (using PDI) | Volume | Hadoop (MapReduce) | Distributed/ Parallel |
| [50] | structured (Sensor Data), Semi-struct. (IoT data) | MongoDB collections (reports, visualization) | Collect, Clean, Normalize, Aggregate | Processing (keras, jupyter), visualization (Knowage) | Variety | MongoDB (storage), kafka (streaming) | Distributed |
| [51] | Un-struct. (UGC text), product retailers, and reviews | NoSQL DW | Data extraction, MD concept mapping, Opinion analysis, Schema integration | Yes (TOS, mongoDB) | Volume, Variety | - | - |

proposal ensures accuracy and reduces disk overhead while preventing any loss of stream data.

In [41], Mallek et al. created BigDimETL, an ETL method for moving large volume of data from Hbase (a NoSQL column-oriented database) to a target DW. It uses the MapReduce framework and focuses on selection and projection operations. Afterward, they expanded their work in [42, 43] by using column-oriented tables to create a multidimensional (MD) structure from input data during the extraction step and by incorporating the join operation in the transformation step. Later, in [44], they defined a conversion operation algorithm that migrates semi-structured documents to a column-oriented structure during the ETL extraction step. This algorithm was implemented as a new TOS-Big Data component for converting JSON data to HBase databases.

Similarly, the "Web ETL" framework defined in [45] also uses MapReduce to transform and load web pages. This work involves filtering techniques to remove duplicate records and a transformation algorithm composed of three steps: tokenize fields, sort DB, and match records. The data is then transferred into a distributed Hadoop-based DW during the loading step. However, the multidimensional structure of the DW needs to be presented in this work. Authors in [46] developed an on-demand ETL architecture for data analysis, focusing on dispersing data across multiple collections. In [47], they introduced an algorithm to detect identifiers and references in multiple document stores. Nevertheless, there is still a need to formalize ETL operations. Unlike other methods, this architecture involves user interaction during the data integration.

In [48], the author made an ETL framework for Big Data that has three layers: an ETL workflow designer layer (an open-source ETL framework), a middle layer with a UDF component, a recommender, cost model, and monitoring agent; and a distributed framework layer for running UDFs in parallel (for example, Hadoop). Authors in [49] also created a cp-UDF generator that is easy to add to Pentaho data integrator (PDI). It makes creating efficient parallel configurations for complex ETL tasks on large datasets possible. The sentiment analysis workflow for product reviews demonstrated the framework's effectiveness but did not provide details on the cleaning step.

Reference in [50] introduces a building management system (BMC) that utilizes ETL processing and data integration to enhance operational efficiency and building performance. It gathers data from sensors, IoT devices, and building systems, processes it using ML and AI tools, and loads it into scalable storage for real-time adjustments. The authors proposed MATRYCS, a scalable architecture that enables comprehensive data integration from various sources for informed decision-making. However, challenges include managing data complexity, ensuring consistency across sources, and adapting to evolving data formats and integration requirements.

Besides, Moalla et al. [51] conducted a study on sentiment analysis to help decision-makers analyze opinions from user-generated content (UGC) on social media platforms such as Facebook, Twitter, and YouTube. In their plan, data would be extracted and cleaned, social media data mart schemas would be modeled, and these schemas would be turned into a generic data warehouse schema based on semantic relationships. The UGC opinion analysis in this process uses a supervised learning classification method described in [52]. Although the experimental results were consistent, more formalization and design are needed to further explore the ETL process steps.

### 6.1.2 Comparison of technical Big Data ETL approaches

Several ETL approaches (discussed in Sect. 6.1.1) define efficient ETL solutions for Big Data integration. They utilize big data frameworks and paradigms for managing NoSQL database systems, parallel and distributed processing, and other big data-specific technologies. Table 3 categorizes and compares these approaches based on the criteria defined in Sect. 4. Most focus on different data source formats, such as JSON or XML documents, raw data, databases, and UGC. However, some of these approaches need to specify the multidimensional structure of the data warehouse, for instance [40, 45].

Big Data ETL approaches are designed to efficiently handle specific operations and tasks such as filter, tokenize, join multiple documents, and analyze opinions. ETL tools like Talend Open Studio (TOS) for Big Data, Pentaho data integration (PDI), and Apache Kafka are used to test and validate these operations. Talend is one of the most widely used ETL tools due to its exceptional qualities, such as being open, innovative, and robust. It also offers data quality, an open profiler, an integrated product, and on-demand services, making it a popular choice among data integration tools [53].

Our research shows that MapReduce is an effective solution for ETL processes when dealing with large amounts of data. Researchers of [42–45, 49] have conducted studies that demonstrate that MapReduce is a widely adopted programming model and processing technique for handling extensive data processing tasks on a distributed cluster of computers. This model is designed to manage parallel processing efficiently and ensure fault tolerance, making it highly suitable for processing large datasets.

Apache Hadoop is a widespread implementation of the MapReduce paradigm, which analyzes large data sets. Other frameworks, like Apache Spark, offer additional features. Hadoop Distributed File System (HDFS) is an essential component of the Apache Hadoop framework for managing large data sets across multiple machines. It is also used for analyzing large volumes of data.

The main focus of the approaches listed in Table 3 is to use various technologies and tools to acquire, process, store, and analyze large datasets. Most of the presented works concentrate on the physical implementation of Big Data integration processes by offering ETL algorithmic solutions. However, it is essential to emphasize the significance of the modeling aspect, which can clearly represent the data warehousing and ETL steps at a conceptual level.

**Table 4** Comparison of Big Data ETL modeling approaches (specific and UML)

| | Modeling language | Meta-model | Modeling level | | | Visual. Flow | Data Sources | Target | ETL models | ETL implement. | Big Data Vs | Big Data Technology | Architecture |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Approaches | | | C | L | P | | | | | | | | |
| [69] | data pipelines | × | × | | | Data flow | Un-struct (human/machine) | label data | collect, ingest, process | - | Velocity, Variety | - | Distributed |
| [72] | specific | | × | | × | Process | Struct., Semi-struct., Un-struct. | generic data model | classifier, mapping | yes (oracle) | Volume, Variety | - | Distributed |
| [54] | specific notation | | | × | | Data flow | Semi-struct. | - | data partition, mapper, reducer | yes | Volume | MapReduce | Distributed |
| [55] | specific notation | | × | | | Data flow | Semi-struct. | - | transform fact & dimension | yes (TOS-BD) | Volume | - | - |
| [56] | UML deployment diagram | | × | | | Data flow | Struct., Semi-struct., Unstruct. | - | CDC, SK, DQV, SCD | yes (TOS-BD) | Variety | Hadoop | Parallel/ Distributed |
| [67] | UML deployment diagram | | × | | | Data flow | Struct., Semi-struct., Unstruct. | DW HDFS | Sqoop, flume, data click | - | Variety | Hadoop | Parallel/ Distributed |
| [68] | UML class diagram | | × | | | - | Struct. | - | CDC | - | Velocity | Spark, Kafka | Parallel/ Distributed |
| [38] | UML class diagram | × | × | × | × | Data flow | Semi-Struct (twitter, web) | MD schema | clean, convert, load tweets | yes (oracle TOS, Python) | Variety, Volume, Velocity | - | - |
| [70, 71] | UML activity diagram | | × | | | Data/ control flows | Semi-struct. (twitter) | DW (column-oriented DB) | select, project, join, convert | yes (using TOS-BD) | volume | MapReduce | Parallel/ Distributed |

Table 5 Comparison of Big Data ETL modeling approaches (based on BPMN)

| | Modeling language | Meta-model | Modeling level | | | Visual. Flow | Data Sources | Target | ETL models | ETL implement. | Big Data Vs | Big Data Technology | Architecture |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Approaches | | | C | L | P | | | | | | | | |
| [57] | BPMN | - | × | | | Process/Data flows | Semi-Struct (TPC-DS) | DW (MongoDB) | Transform fact and dimensions | yes (using TOS-BD) | Volume | - | - |
| [58] | BPMN | - | × | | | Data flow | Unstruct. (UGC text) | SDW (Sentiment dim.) | Extract, Analyze sentiment | - | - | - | - |
| [65] | BPMN | × | × | | | Process, Operation, Data model | Unstruct. (UGC text) | Social DW | standard/semantic mapping | ETL4Social-Tool (TOS-BD) | Variety | - | - |
| [66] | BPMN | - | × | | | Process/Data flows | Unstruct. (UGC text) | SDW (Topic dim.) | Crawl, preprocess, detect topic | ETL4Social-Topic (TOS-BD) | Variety | - | - |
| [36] | BPMN | - | × | | | Process/Data flows | Unstruct. (UGC text) | SDW (Sentiment dim.) | Crawl, preprocess, sentiment | ETL4Social-Sentiment (TOS-BD) | Variety | - | - |

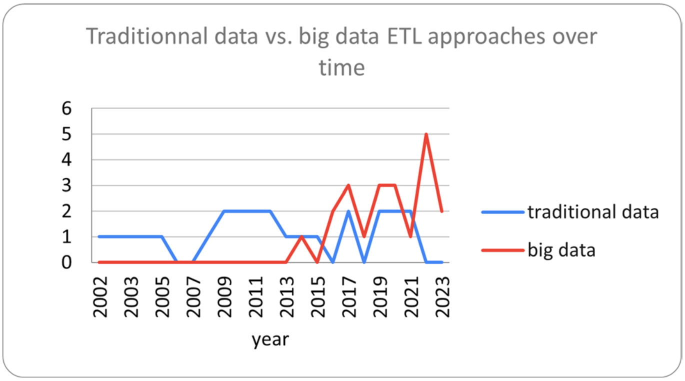


**Fig. 6** Evolution of ETL approaches

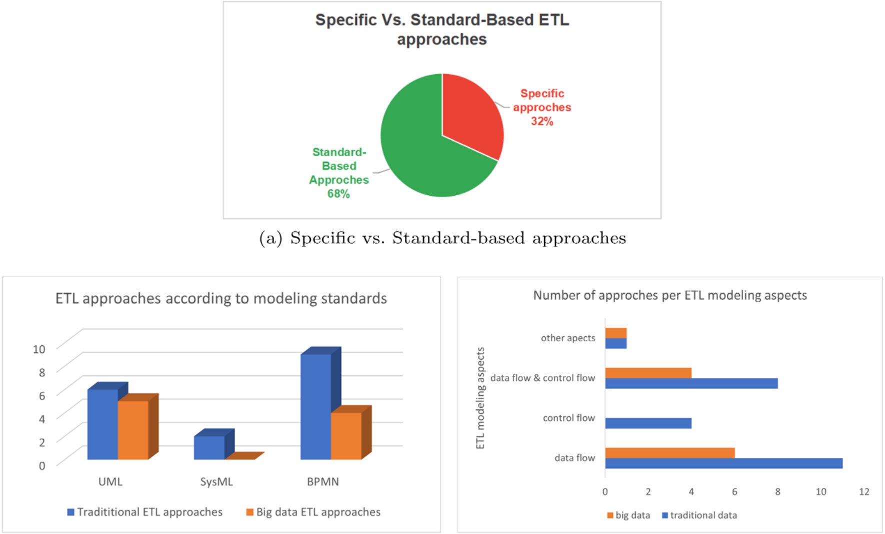
## 6.2 ETL modeling for Big Data integration

Several researchers have devoted their attention to conceptual design of Big Data integration steps, providing comprehensive models for the multidimensional structure of Big Data warehousing and ETL processes and operations.

### 6.2.1 Summary of Big Data ETL modeling approaches

In [54], authors presented a five-step process for a distributed ETL platform utilizing the MapReduce (MR) paradigm. In [55], they introduced specific graphical notations for modeling ETL phases: partitioning data sources, transforming them through the Map phase, and merging them through the Reduce phase. Later, they optimized ETL performance by parallelizing at two levels and introducing a fine-grained parallel/distributed ETL approach [56]. It should be noted that their research solely concentrated on the volume of data.

Authors in [57] used business process modeling and notation (BPMN) to define transformation rules that turn the DW multidimensional schema into a document-oriented model that works with MongoDB. Similarly, BPMN was also utilized in [58] to design a process that extracts Facebook comments and converts them into user opinions to detect the sentiment (positive, negative or neutral). Several studies from the past 10 years have recommended incorporating user opinions as an analysis dimension in the DW [59–61]. However, considering the topic of interest is vital to efficiently analyzing user opinions on social media [62–64]. In [65, 66], Walha et al. proposed a BPMN process model that transforms informal UGC texts into valid topic and sentiment dimensions in the DW. This model was implemented on TOS for Big Data, which provides customized jobs that automate a significant part of this complex process and make the work of the ETL designer easier.



(a) Specific vs. Standard-based approaches

(b) UML, SysML, and BPMN approaches

(c) ETL approaches modeling aspects

**Fig. 7** Classification of ETL modeling approaches

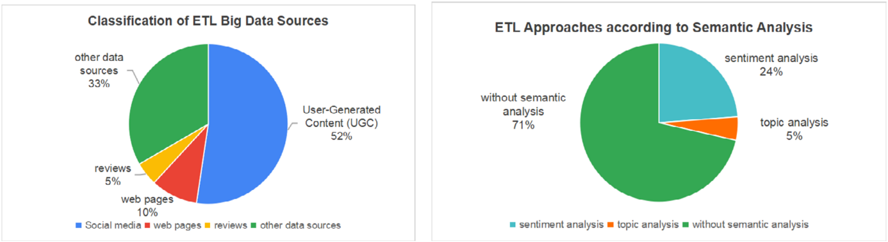
Some researchers have used UML language to design ETL for integrating Big Data. For example, authors proposed a model in [67] for extracting Big Data, incorporating new stereotypes and ETL methodology in data warehouses. Case studies were conducted using three commonly used tools in the extraction process: sqoop, flume, and data click. Another study [68] suggested utilizing UML diagrams for Batch and Stream processing strategies in the ETL process as part of the DOD-ETL framework, which aims to improve the bottleneck of near-real-time ETL by executing CDC functionality. Later, Raj et al. [69] developed a conceptual method for modeling data pipelines, utilizing ETL/ELT transformations, multiple applications, and different data types. The study focuses on both human-generated and machine-generated data sources. The model automates monitoring, fault detection, mitigation, and alarming at various data pipeline steps, aiming to solve data management challenges with minimal human intervention. Recently, Mallek et al. presented two papers [70, 71], respectively, outlining an abstract view of the ETL extraction and transformation phases. The authors proposed conceptual modeling of fundamental ETL operations (e.g., Project, Join, and Conversion) through the UML activity diagrams. These operations can be executed in parallel across a large dataset using MapReduce. This approach minimizes ETL time consumption by integrating parallelism into traditional ETL processes.

The study explored different approaches to acquiring, processing, storing, and analyzing Big Data. However, none of the approaches offered a complete architecture for warehousing Big Data, encompassing requirements, implementation strategy, conceptual design, and ETL phases. In response to this, in [36], authors proposed a conceptual framework known as ETL4Social, which expertly tackles the complexity of the ETL process when integrating user-generated content (UGC) into data warehouse (DW) design. The authors partitioned the ETL modeling aspects into processes, operations, and data and introduced an architecture comprising three layers: meta-modeling, modeling, and instantiation. A meta-model was proposed for each aspect to organize the applicable concepts and relationships in various contexts and social media. The accuracy of the models was demonstrated through an illustrative example that mapped generic models of UGC into DW. Similarly, authors in [37] introduced a multi-layer model consisting of three levels: Meta-Model Level, which is defined through a UML class diagram that outlines the elements and concepts in data warehousing phases; the Meta-model Specification level, which specifies possible instances of classes linked with "Is A" relations; and conceptual model level, which presents entities as subclasses or instances of classes. The authors analyzed the evolution of COVID-19 on Twitter as a case study to demonstrate the effectiveness of their approach.

To reduce manual effort in integrating heterogeneous data, Pan et al. [72] have recently applied semantic-similarity methods to automate the schema-mapping process. Their research proposed mapping raw data from the building energy domain into a generic data model, improving schema matching accuracy.

**Table 6** Classification of ETL approaches for the main 4Vs and application domains

| ETL Approaches (Big Data) | Volume | Variety | Velocity | Veracity | Application Domains |
|---|---|---|---|---|---|
| Bala et al. [54, 55] | x | | | | Education ministry (student integration) |
| Bala et al. [56] | | x | | | Education ministry (student integration) |
| Mosquera et al. [67] | | x | | | - |
| Yangui et al. [57] | x | | | | - |
| Machado et al. [68] | | | x | | - |
| Walha et al. [36, 65, 66, 73] | | x | | | Opinion analysis (user sentiment and topic) |
| Dhaouadi et al. [37] | x | x | x | | Tweet integration |
| Mallek et al. [42–44, 70, 71] | x | | | | Distribution/ Parallel processing of Tweet Integration |
| Gupta et al. [45] | | x | | | Data transformation to de-redundant the records |
| Souibgui et al. [46, 47] | | x | | | Identifiers and references detection from document stores |
| Raj et al. [69] | | x | x | | Human/machine transformation |
| Moalla et al. [51, 52] | | x | | | Sentiment analysis/opinion detection (social media) |
| Pan et al. [72] | x | x | | | Mapping Energy Data |



(a) Big Data sources

(b) Semantic Analysis for UGC Integration

**Fig. 8** Classification of ETL approaches (Big Data)

### 6.2.2 Comparison of Big Data ETL modeling approaches

Tables 4 and 5 present and compare approaches that tackle the challenge of representing ETL operations' sequence and passage of inputs and outputs in a Big Data context. Some authors, such as [55], use specific notations, while others focus on modeling standards like UML class diagrams (e.g., [37]) or UML deployment diagrams (e.g., [67]) to represent the ETL steps' static aspect. Control flows of ETL activities are designed using UML activity diagrams (e.g., [70, 71]) or BPMN diagrams (e.g., [57, 65, 66]).

A well-organized meta-model encompassing all ETL's essential elements and relationships is valuable for any design approach. Only the works [36, 37] presented meta-models, which design ETL operations and functionalities for Big Data. However, the authors have yet to consider parallelism and distribution aspects in their modeling.

The involvement of decision-makers in the design of data warehousing (DW) and extract, transform, and load (ETL) processes is crucial. According to [47], there is research on active user interaction during the big-data integration process. Similarly, authors in [65] involved the designer in detecting topics from user-generated texts and validating the topic dimension schema. Due to the various constraints of social media (such as data variety), some ETL operations and tasks can be customized to meet the designer's specific requirements. To provide more flexibility in the ETL process, [49] proposed user-defined functions.

Recent approaches in Table 3 integrate Big Data extraction and transformation techniques into reusable conceptual models, but some need to focus more on ETL modeling aspects. Firstly, separating sequences and data flows can simplify the ETL process's complexity. Secondly, a conceptual framework organizing essential concepts on meta-models, such as the works in [36] and [37], can provide a generic approach to ETL modeling. Thirdly, according to some researches, including [47] and [65], the designer's involvement in modeling and ETL processes is also crucial. Customized ETL processes and tasks ensure more flexibility during the big-data integration process. Finally, the current literature must provide a clearer understanding of how to effectively scale conceptual models to address the increasing complexity, heterogeneity, and volume of Big Data in ETL workflows.

Tables 3, 4 and 5 summarize the ETL processes for selecting, filtering, and normalizing Big Data sources for relevant decision-making. Most approaches have focused on the extraction (E) step, as it is the most complex step of the ETL scenario (e.g., [44, 67, 70, 71]). However, other researchers have also concentrated on the transformation (T) step, such as [40, 42, 44, 45, 49, 57, 65]. Only the work [37] covers all ETL steps.

## 7 Discussion and future directions

### 7.1 Discussion and findings

For an effective DSS, it is crucial to have a well-designed ETL process that can cater to the data sources, data warehouses, and environments where data is converted. In this section, we will discuss the benefits and limitations associated with different aspects of the ETL process examined in this paper.

#### 7.1.1 From traditional to Big Data ETL approaches

Traditional ETL approaches studied in this paper (Sect. 5) simplify the process of designing workflows for the extraction, transformation, and loading of transactional data. Most approaches propose conceptual frameworks that specify valid ETL workflows at several levels. They propose reusable and flexible concepts, models, and meta-models to transform structured data into a Data Warehouse. These proposals are helpful for business intelligence systems defined with limited and structured data sources stored in transactional databases. However, with the rise of Big Data stored in NoSQL databases, traditional ETL workflows, activities, concepts, models, frameworks, and techniques are not suitable to handle the vast amounts of various and evolutional semi-structured and unstructured data.

Recently, designing the ETL process for Big Data is challenging for several researchers due to the growth of schema-less databases. This matter makes data analysis and integration more difficult. Some works (detailed in Sect. 6.1) have focused on applying various technologies and tools for Big Data acquisition, processing, storage, and analysis. However, most approaches concentrate on the physical implementation of the ETL algorithms without taking advantage of their conceptual level. This solution can make it challenging to represent, analyze, and understand complex ETL workflows and provide support for automatic code generations. Several researchers (discussed in Sect. 6.2) have focused on modeling the Big Data integration process (ETL tasks and operators) and multidimensional models for big data warehouses.

This survey explores the development of the ETL approaches from 2000 to the 2023. Figure 6 shows the number of approaches that have emerged over the years. Most traditional approaches discussed in this paper were developed between 2009–2012 and 2019–2021. However, ETLs for Big Data, which appeared in 2014, have evolved rapidly due to the growing usage of unstructured and semi-structured data.

### 7.1.2 Specific versus standard ETL modeling approaches

Irrespective of the data type, ETL modeling approaches can be classified into two main groups: Specific and Standard. Specific approaches use formal and conceptual notation, whereas standard approaches use common languages such as UML, BPMN, and SysML. As depicted in Fig. 7a, specific approaches only comprise 32% of the total ETL approaches because designers must be familiar with specific notation. Conversely, 68% of ETL approaches use standards, providing generic and more readily understandable models. This point reduces designer effort, cost, and time. Both specific and standard approaches have unique strengths, and designers can choose the best approach based on their needs and requirements.

### 7.1.3 Modeling standards of ETL approaches

In Fig. 7b, the ETL approaches are classified based on whether they adhere to modeling standards or notations. Our analysis indicates that BPMN and UML activity diagrams are suitable for designing complex control flows within an ETL process. While these two standards share similarities, BPMN-based approaches provide a modeling structure that closely resembles the ETL process by considering the dynamic nature of connections between processes. Therefore, BPMN is the best choice for designing and validating the ETL process. Indeed, this rating provides control objects that allow ETL designers to merge data and control flows with the same tool. In addition, it ensures the concrete specification of an ETL process. From an implementation efficiency point of view, it covers a communication gap that can sometimes appear between the design and implementation of a business process.

### 7.1.4 Modeling aspects of ETL approaches

When examining the modeling aspects addressed by conceptual ETL modeling approaches, as shown in Fig. 7c, it is evident that most ETL approaches concentrate primarily on data flow design. In contrast, others strive to simplify the complexity of ETL design by separating data and control flows. Some ETL approaches (e.g., [12, 36]) have also introduced new modeling aspects, such as templates, data models, and expressions.

Our study suggests that separating modeling aspects is essential for creating a simpler, more dynamic, and more flexible approach. ETL designers and practitioners should prioritize this separation, particularly in Big Data environments where new specific concepts are being defined.

### 7.1.5 ETL approaches according to Big Data Vs

Table 6 categorizes the different ETL approaches based on various features of Big Data. It is worth noting that most studies on NoSQL data integration only consider the volume or variety of the data. However, there have been some advancements in addressing the speed of data streams, known as velocity. Additionally, it would be preferable to expand the Big Data integration approaches to include other Big Data features like veracity and variability.

### 7.1.6 Social media analytics for ETL approaches

With the increasing use of social networks as a significant data source, recent approaches (approximately 24%) take advantage of user-generated content (UGC) on social media to enhance business intelligence processes (8a). Researchers such as [36, 60, 61] have effectively incorporated valuable insights from UGCs (such as tweets) into new dimensions and measures in the data warehouse, providing adapted ETL tasks for user opinions analysis.

In recent research (71%), such as the work by Mallek [70], there has been a focus on modeling the primary phases of Big Data ETL for handling large volumes of data by adapting by adapting ETL processes to distributed processing. Nonetheless, there is a demand for a more comprehensive analytical and semantic analysis of user interactions, particularly when dealing with heterogeneous and informal data, to detect and integrate opinions. Other approaches have proposed ETL modeling solutions for sentiment analysis (e.g., [52, 66]) and topic analysis [65] of UGC text.

Figure 8b reveals that only 29% of the surveyed approaches have addressed these issues. Recent studies, as [51, 74, 75], have underlined a growing interest in using natural language processing (NLP) and machine learning (ML) to transform user-generated content text into valuable insights. It is highly recommended that ETL designers consider integrating cleaning, normalization, and sentiment analysis algorithms into the ETL process.

## 7.2 Current and future trends on Extract–Transform–Loading processes

Figure 5 shows that research activities and publications for transactional systems have steadily decreased from 2017 to 2023. In contrast, there has been an increase in ETL approaches for handling Big Data during the same period. This shift reflects the focus of modern organizations on integrating unstructured (e.g., multimedia data) and semi-structured data in addition to the structured data typically managed by transactional systems. The use of new devices such as mobile devices, smartphones, IoT devices, and satellites has introduced new formats of data, including social media data (comments, user-generated data, posts, videos, texts), web content (blogs, forums, web pages), customer service chat logs, flat files, NoSQL databases, clickstream data, apps, reviews, and product purchases.

### 7.2.1 Trends on Big Data techniques and practices

With the emergence of Big Data sources, new architectures have been proposed, posing challenges for integration, especially in processing and storage. Detailed current and future trends are available for each of these challenges.

***Challenge 1: ETL complexity with growth in data volume***

In recent years, researchers and practitioners have utilized various techniques, architectures, storage, and processing systems to create efficient ETL workflows that can handle large volumes of data. Some of the approaches used include:

- The widespread use of distributed processing frameworks such as Apache Spark to enable in-memory processing capabilities and support transformations on data stored in NoSQL databases [76, 77]. Optimizing ETL workflows for parallel processing is the current solution. One efficient approach is to utilize cloud-based ETL services like AWS Glue, Azure Data Factory, or Google Cloud Platform to scale on demand [78, 79].
- Modern data architectures, including Data Lakes, NoSQL databases, cloud data warehouses, Apache Hadoop Distributed File System (HDFS), are adopted for loading transformed data to target systems. The trend toward data lakehouse architectures, which combine data lakes' flexibility with data warehouses' performance, is also gaining traction [80, 81].
- Transition from traditional ETL to ELT (Extract–Load–Transform), where data is first loaded into a data lake or data warehouse and then transformed in place. This approach is more aligned with modern, scalable data architectures and leverages the power of cloud-based systems to efficiently handle large-scale transformations after loading [82].

***Challenge 2: ETL processing with Big Data variety***

Big Data involves dealing with a wide variety of data types and sources. Managing this variety is crucial in the ETL process to ensure accurate and efficient processing and integration into the target system. One of the primary challenges is the heterogeneity of data sources. NoSQL databases like MongoDB, Cassandra, and Hadoop HDFS are designed to handle unstructured and semi-structured data, providing flexibility in storage and retrieval [83].

Unstructured data, such as text documents, emails, social media posts, videos, images, and sensor data, is increasingly prevalent in modern data-driven environments. Dealing with this data type necessitates advanced ETL processing to handle and integrate diverse data formats. Traditional ETL tools face difficulties with unstructured data, leading to the development of new approaches that focus on current trends:

- Advanced techniques for gathering unstructured data, such as web scraping, API crawling, and file parsing. Tools like Apache NiFi, Talend, and Informatica can streamline the extraction process from diverse sources [53].
- Advanced natural language processing (NLP) techniques and semantic analysis tools to comprehend and interpret data [84], techniques like extracting entities,

relationships, and sentiments to transform user-generated text into topics [65] and sentiments [66], or knowledge graphs and ontologies to provide context and meaning to the data [85].

- *Text analytics and sentiment analysis algorithms* analyze text data to extract sentiments, trends, and patterns, converting unstructured text into actionable insights [75].
- Machine or deep learning models and AI techniques are increasingly utilized to automate the classification, tagging, and extracting relevant information from unstructured data, thereby enhancing the ETL process [50, 51, 86].
- Large language models (LLMs) such as OpenAI's GPT-4, BERT, RoBERTa, and T5 due to their significant capabilities for understanding and generating human language [87]. These models are particularly useful in data processing and analysis within key domains such as customer support, healthcare, and intelligent transportation systems.

Using advanced tools, databases, and algorithms, DSS effectively manages Big Data variety and heterogeneity, specifically unstructured data, enabling comprehensive analysis and informed decision-making.

***Challenge 3: ETL processing with Big Data velocity***

Modern organizations place great importance on the speed of processing and analyzing data in the data integration process. One challenge of ETL approaches for Big Data is handling high-speed data streams in real-time or near-real-time to generate timely insights. Researchers should focus more on velocity with systems that handle continuous data streams, like social media feeds, sensor data, or financial market data, and with applications that require immediate insights or decision-making based on incoming data, such as fraud detection, IoT analytics, customer recommendations, or automated trading systems.

Modern data pipelines may include both batch and streaming data processing to address various use cases within an organization [88]. It requires advanced architecture and tools to handle continuous data flow and processing when combining ETL processes with real-time data streaming. Notable technologies associated with high-velocity data processing are Apache Kafka, Apache Flink, Apache Storm, and Apache Spark Streaming [40].

***Challenge 4: ETL processing with Big Data veracity and integrity***

The veracity pertains to the reliability of uncertain data. It encompasses the data's truthfulness, accuracy, precision, and correctness [4]. As stated in [89], high-quality data aligns with reality and is easily accessible, up-to-date, non-repetitive, and exhaustive.

The data on which ETL (extract, transform, and load) tasks are performed originates from diverse sources containing data of varying quality. These sources use inconsistent representations, codes, and formats that must be reconciled. Ensuring data accuracy in ETL processes is crucial for maintaining the reliability and trustworthiness of data in a data warehouse. Organizations can improve data quality and obtain valuable insights from their analytics efforts by implementing robust validation, cleansing, and monitoring techniques. Future strategies to address this challenge may involve:

- Data lakehouse are expected to be a significant area of research in future data management, offering improved data quality, governance, reduced redundancy, and efficient use of time with a simplified schema [90].
- Using machine learning algorithms to detect outliers and anomalies in data that may indicate potential integrity issues [91]
- leveraging blockchain technology, organizations can significantly improve their ETL processes' integrity, security, and traceability, ensuring that data remains accurate, reliable, and compliant with regulatory standards [92].

Advanced techniques, tools, scalable storage solutions, and robust processing frameworks enable ETL processes to effectively manage and extract insights from Big Data. However, it is not just about technical solutions. Current approaches need to emphasize the importance of data modeling to ensure consistency, efficiency, and data integrity. The main question we are tackling is how designers can integrate this technical solution into modeling ETL processes.

### 7.2.2 Trends on ETL modeling

Integrating robust data modeling practices into ETL processes is crucial for building resilient and efficient decision support systems. It is essential to consider modeling challenges such as developing automated ETL solutions, standardized modeling, and cost-effective approaches. New trends include:

- Standardizing the ETL to illustrate ETL workflows and data transformations to enhance stakeholder communication and understanding. As shown in Fig. 7a, 68% are standard-based ETL approaches. Current trends emphasize using BPMN notation to model and automate data workflows, as it can bridge the gap between business process modeling and ETL design, ensuring that data processes align with business objectives (Fig. 7b).
- Aligning with advanced algorithms for Big Data integrations (e.g., ML and DL models, semantic analysis, NLP, and pre-processing techniques) presents a promising future for ETL processes. This alignment will allow for the development of components, modules, or pipelines that can be reused across multiple projects, datasets, or use cases, enhancing the reusability of the ETL process for Big Data.
- Efforts on redefining concepts to align with extensive data features, including information about data sources, target schemas, transformation rules, data lineage, and quality rules. By abstracting transformation logic into metadata configurations, organizations can more easily adjust to changing data sources and requirements.
- A meta-model is required to encapsulate the structure and relationships between ETL processes, functionalities, and analytical models within the DSS framework. Specify metadata definitions, data lineage, and transformation rules that govern how data flows through the system, ensuring consistency and traceability from extraction to analysis.

- New trends involve implementing strategies for schema evolution, using flexible data models, and adopting agile ETL development methodologies to adapt ETL changes in data schemas and structures without disrupting data flow.

## 8 Conclusion

The extract, transform, and load process retrieves data from different sources, modifies it to meet specific analytical needs, and stores the processed data in a dedicated storage system called a data warehouse. A well-designed ETL process is crucial for all DSS development projects as it is closely linked to its success and ease of maintenance. In the early 2000s, researchers focused on modeling ETL processes, resulting in numerous studies published in this area, each with advantages. Nevertheless, Big Data has presented new challenges.

This paper comprehensively investigates different ETL approaches from the past 2 decades. The approaches are categorized into specific and standard-based approaches for traditional data and technical and modeling approaches for Big Data. The past 10+ years have been dedicated to exploring and implementing efficient ways to manage the complexities of Big Data due to increasing volume, speed, heterogeneity, and variety of data sources. This feature has involved blending various processing and storage methods and applying them in real-world scenarios. Additionally, recent trends in Big Data integration are explored, and potential developments and modeling expected to significantly impact the next generation of ETL and data integration technology are discussed.

Overall, this survey has contributed to a deeper understanding of data integration issues, offering a comprehensive overview of key trends, challenges, and opportunities for researchers and data practitioners to enhance strategies, policies, and practices. Several critical perspectives emerge from this study: The survey provides insights into data integration issues and solutions. One crucial aspect of DSS is the increasing popularity of exploratory data analysis, which requires thorough exploration and integration of Big Data. Advanced AI systems, such as LLM models, introduce new challenges with automated data exploration. Further research is needed to understand the underlying causes of the knowledge acquired through exploratory data analysis. Recent studies (e.g., [93]) advocate for a transparent and explicable approach to data analysis, aiming to provide qualitative and quantitative explanations. Our future research will focus on a comprehensive study of approaches that align with exploratory data analysis.

**Author contributions** Authors collaborated in writing the manuscript.

**Data availability** No datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

1. Dhaouadi A, Bousselmi K, Gammoudi MM, Monnet S, Hammoudi S (2023) Data warehousing process modeling from classical approaches to new trends: main features and comparisons. Data 7(8):113
2. Demarest M (1997) The politics of data warehousing. June, http://www.hevanet.com/demarest/marc/dwpol.html, 6(03), 1998
3. Nwokeji JC, Matovu R (2021) A systematic literature review on Big Data extraction, transformation and loading (ETL). In: Intelligent Computing: Proceedings of the 2021 Computing Conference, vol 2. Springer International Publishing, pp 308-324
4. Vassakis K, Petrakis E, Kopanakis I (2018) Big data analytics: applications, prospects and challenges. A roadmap from models to technologies, Mobile big data, pp 3–20
5. Petersen K, Vakkalanka S, Kuzniarz L (2015) Guidelines for conducting systematic mapping studies in software engineering: an update. Inf Softw Technol 64:1–18
6. Vassiliadis P, Vagena Z, Skiadopoulos S, Karayannidis N, Sellis T (2001) ARKTOS: towards the modeling, design, control and execution of ETL processes. Inf Syst 26(8):537–561
7. Vassiliadis P, Simitsis A, Skiadopoulos S (2002) Conceptual modeling for ETL processes. In: International Workshop on Data Warehousing and OLAP. ACM, pp 14–21
8. Vassiliadis P, Simitsis A, Georgantas P, Terrovitis M, Skiadopoulos S (2005) A generic and customizable framework for the design of ETL scenarios. Inf Syst 30(7):492–525
9. Vassiliadis P, Simitsis A, Baikousi E (2009) A taxonomy of ETL activities. In: International Workshop on Data Warehousing and OLAP (DOLAP). ACM, pp 25–32
10. Köppen V, Brüggemann B, Berendt B (2011) Designing data integration: the ETL pattern approach. UPGRADE Eur J Inform Prof 3:49–55
11. El-Sappagh SHA, Hendawi AMA, El Bastawissy AH (2011) A proposed model for data warehouse ETL processes. J King Saud Univ-Comput Inf Sci 23(2):91–104
12. Petrović M, Vučković M, Turajlić N, Babarogić S, Aničić N, Marjanović Z (2017) Automating ETL processes using the domain-specific modeling approach. Inf Syst e-Bus Manag 15:425–460
13. Deme A, Buchmann R (2021) A technology-specific modeling method for data ETL processes. In: AMCIS
14. Oliveira B, Belo O (2016) An ontology for describing ETL patterns behavior. In: 5th International Conference on Data Management Technologies and Applications, pp 102–109
15. Oliveira B, Belo O (2017) Approaching ETL processes specification using a pattern-based ontology. In: Data Management Technologies and Applications; Communications in Computer and Information Science, vol 737. Springer, pp 65–78
16. Jacobson L, Booch JRG (2021) The unified modeling language reference manual
17. Trujillo J, Luján-Mora S (2003) A UML based approach for modeling ETL processes in data warehouses. In: International Conference on Conceptual Modeling. Springer Berlin Heidelberg, pp 307–320
18. Luján-Mora S, Vassiliadis P, TrujilloJ (2004) Data mapping diagrams for data warehouse design with UML. In: International Conference on Conceptual Modeling. Springer Berlin Heidelberg, pp 191-204
19. Song X, Yan X, Yang L (2009) Design ETL metamodel based on UML profile. In: International Symposium on Knowledge Acquisition and Modeling, vol 3. IEEE, pp 69–72
20. Muñoz L, Mazón, JN, Pardillo J, Trujillo J (2008) Modelling ETL processes of data warehouses with UML activity diagrams. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer Berlin Heidelberg, pp 44–53
21. Muñoz L, Mazón JN, Trujillo J (2010) A family of experiments to validate measures for UML activity diagrams of ETL processes in data warehouses. Inf Softw Technol 52(11):1188–1203
22. Mallek H, Walha A, Ghozzi F, Gargouri F (2014) ETL-web process modeling. In: Advances on Decisional Systems Conference (ASD)
23. Biswas N, Chattopadhyay S, Mahapatra G, Chatterjee S, Mondal K C (2017) SysML based conceptual ETL process modeling. In: Computational Intelligence, Communications, and Business Analytics International Conference (CICBA). Springer Singapore, pp 242–255
24. Friedenthal S, Moore A, Steiner R (2014) A practical guide to SysML: the systems modeling language. Morgan Kaufmann

25. Biswas N, Chattapadhyay S, Mahapatra G, Chatterjee S, Mondal KC (2019) A new approach for conceptual extraction-transformation-loading process modeling. Int J Amb Comput Intell (IJACI) 10(1):30–45
26. Chinosi M, Trombetta A (2012) BPMN: an introduction to the standard. Comput Stand Interfaces 34(1):124–134
27. Wilkinson K, Simitsis A, Castellanos M, Dayal U (2010) Leveraging business process models for ETL design. In: International Conference on Conceptual Modeling. Springer Berlin Heidelberg, pp 15–30
28. Nabli A, Bouaziz S, Yangui R, Gargouri F (2015) Two-ETL phases for data warehouse creation: design and implementation. In: Advances in Databases and Information Systems: East European Conference (ADBIS). Springer, pp 138–150
29. El Akkaoui Z, Zimányi E (2009) Defining ETL worfklows using BPMN and BPEL. In: International workshop on Data warehousing and OLAP. pp 41–48
30. El Akkaoui Z, Mazón JN, Vaisman A, Zimányi E, (2012) BPMN-based conceptual modeling of ETL processes. In: Data Warehousing and Knowledge Discovery (DaWaK, (2012). Springer, Berlin Heidelberg, pp 1–14
31. El Akkaoui Z, Zimányi E, Mazón JN, Trujillo J (2013) A BPMN-based design and maintenance framework for ETL processes. In J Data Warehous Min (IJDWM) 9(3):46–72
32. El Akkaoui Z, Vaisman AA, Zimányi E (2019) A quality-based ETL design evaluation framework. ICEIS 1:249–257
33. Oliveira B, Oliveira Ó, Belo O (2021) Using BPMN for ETL conceptual modelling: a case study. In: Data, pp 267–274
34. Awiti J, Vaisman AA, Zimányi E (2020) Design and implementation of ETL processes using BPMN and relational algebra. Data Knowl Eng 129:101–837
35. Oliveira B, Belo O (2012) BPMN patterns for ETL conceptual modelling and validation. In: Foundations of Intelligent Systems International Symposium (ISMIS (2012). Springer, Berlin Heidelberg, pp 445–454
36. Walha A, Ghozzi F, Gargouri F (2019) From user generated content to social data warehouse: processes, operations and data modelling. Int J Web Eng Technol 14(3):203–230
37. Dhaouadi A, Bousselmi K, Monnet S, Gammoudi MM, Hammoudi S (2022) A multi-layer modeling for the generation of new architectures for big data warehousing. In: International Conference on Advanced Information Networking and Applications. Springer, pp 204–218
38. Iribarne L, Asensio JA, Padilla N, Criado J (2017) Modeling Big data-based systems through ontological trading. Softw Pract Exp 47(11):1561–1596
39. Sahiet D, Asanka PD (2015) ETL framework design for NoSQL databases in dataware housing. Int. J. Res. Comput. Appl. Rob. 3:67–75
40. Mehmood E, Anees T (2022) Distributed real-time ETL architecture for unstructured big data. Knowl Inf Syst 64(12):419–3445
41. Mallek H, Ghozzi F, Teste O, Gargouri F (2017) BigDimETL: ETL for multidimensional big data. In: International Conference on Intelligent Systems Design and Applications (ISDA 2016). Springer, pp 935-944
42. Mallek H, Ghozzi F, Teste O, Gargouri F (2018) BigDimETL with NoSQL database. Procedia Comput Sci 126:798–807
43. Mallek H, Ghozzi F, Gargouri F (2020) Towards extract-transform-load operations in a big data context. Int J Sociotechnol Knowl Dev (IJSKD) 12(2):77–95
44. Mallek H, Ghozzi F, Gargouri F (2022) Conversion operation: from semi-structured collection of documents to column-oriented structure. In: International Conference on Hybrid Intelligent Systems. Springer Nature Switzerland, Cham, pp 585–594
45. Gupta G, Kumar N, Chhabra I (2020) Optimised transformation algorithm for hadoop data loading in web ETL framework. EAI Endorsed Trans Scalable Inf Syst 7(25):e6–e6
46. Souibgui M, Atigui F, Yahia SB, Si-Said Cherfi S (2020) Business intelligence and analytics: on-demand ETL over document stores. In: Research Challenges in Information Science (RCIS 2020). Springer, pp 556–561
47. Souibgui M, Atigui F, Yahia SB, Cherfi SSS (2022) An embedding driven approach to automatically detect identifiers and references in document stores. Data Knowl Eng 139:102003
48. Ali SMF (2018) Next-generation ETL framework to address the challenges posed by big data. In: DOLAP

49. Ali SMF, Mey J, Thiele M (2019) Parallelizing user-defined functions in the ETL workflow using orchestration style sheets. Int J Appl Math Comput Sci 29(1):69–79
50. Pau M, Kapsalis P, Pan Z, Korbakis G, Pellegrino D, Monti A (2022) MATRYCS-a big data architecture for advanced services in the building domain. Energies 15(7):2568
51. Moalla I, Nabli A, Hammami M (2022) Data warehouse building to support opinion analysis in social media. Soc Netw Anal Min 12(1):123
52. Moalla I, Nabli A, Hammami M (2018) Towards opinions analysis method from social media for multidimensional analysis. In: International Conference on Advances in Mobile Computing and Multimedia, pp 8–14
53. Qaiser A, Farooq MU, Mustafa SMN, Abrar N (2023) Comparative analysis of ETL tools in big data analytics. Pak J Eng Technol 6(1):7–12
54. Bala M, Boussaid O, Alimazighi Z (2014) P-ETL: parallel-ETL based on the MapReduce paradigm. In: IEEE/ACS International Conference on Computer Systems and Applications (AICCSA). IEEE, pp 42–49
55. Bala M, Boussaid O, Alimazighi Z (2016) Extracting-transforming-loading modeling approach for big data analytics. Int J Decis Support Syst Technol (IJDSST) 8(4):50–69
56. Bala M, Boussaid O, Alimazighi Z (2017) A fine-grained distribution approach for ETL processes in big data environments. Data Knowl Eng 111:114–136
57. Yangui R, Nabli A, Gargouri F (2017) ETL based framework for NoSQL warehousing. In: Information Systems: 14th European, Mediterranean, and Middle Eastern Conference, (EMCIS). Springer, pp 40–53
58. Walha A, Ghozzi F, Gargouri F (2016) ETL design toward social network opinion analysis. Computer and information science. Springer, Cham, pp 235–249
59. Lanza Cruz IL, Berlanga Llavori R (2018) Defining dynamic indicators for social network analysis: a case study in the automotive domain using Twiter
60. Ben Kraiem M, Alqarni M, Feki J, Ravat F (2020) OLAP operators for social network analysis. Clust Comput 23:2347–2374
61. Moulai H, Drias H (2018) From data warehouse to information warehouse: application to social media. In: International Conference on Learning and Optimization Algorithms: Theory and Applications, pp 1–6
62. Gallinucci E, Golfarelli M, Rizzi S (2015) Advanced topic modeling for social business intelligence. Inf Syst 53:87–106
63. Kurnia PF (2018) Business intelligence model to analyze social media information. Procedia Comput Sci 135:5–14
64. Gutiérrez-Batista K, Campaña JR, Vila MA, Martin-Bautista MJ (2018) Building a contextual dimension for OLAP using textual data from social networks. Expert Syst Appl 93:118–133
65. Walha A, Ghozzi F, Gargouri F (2021) Design and execution of ETL process to build topic dimension from user-generated content. In: International Conference on Research Challenges in Information Science. Springer, pp 374–389
66. Walha A, Ghozzi F, Gargouri F (2024) Extract-transform-load process for recognizing sentiment from user-generated text on social media. In: International Conference on Evaluation of Novel Approaches to Software Engineering. SCITEPRESS, pp 641–648
67. Martinez-Mosquera D, Luján-Mora S, Recalde H (2017) Conceptual modeling of big data extract processes with UML. In: International Conference on Information Systems and Computer Science (INCISCOS). IEEE, pp 207–211
68. Machado GV, Cunha Í, Pereira AC, Oliveira LB (2019) DOD-ETL: distributed on-demand ETL for near real-time business intelligence. J Internet Serv Appl 10:1–15
69. Raj A, Bosch J, Olsson HH, Wang TJ (2020) Modelling data pipelines. In: Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp 13–20
70. Mallek H, Ghozzi F, Gargouri F (2023) Conceptual modeling of big data extraction phase. Int J Hybrid Intell Syst 19(3,4):167–182
71. Mallek H, Ghozzi F, Gargouri F (2023) Conceptual modeling of big data SPJ operations with Twitter social medium. Soc Netw Anal Min 13(1):105
72. Pan Z, Pan G, Monti A (2022) Semantic-similarity-based schema matching for management of building energy data. Energies 15(23):8894
73. Walha A, Ghozzi F, Gargouri F (2017) ETL4Social-data: modeling approach for topic hierarchy. In: KEOD, pp 107–118

74. Hung LP, Alias S (2023) Beyond sentiment analysis: a review of recent trends in text based sentiment analysis and emotion detection. J Adv Comput Intell Intell Inform 27(1):84–95
75. Qi Y, Shabrina Z (2023) Sentiment analysis using Twitter data: a comparative application of lexicon-and machine-learning-based approach. Soc Netw Anal Min 13(1):31
76. Hajji T, Loukili R, El Hassani I, Masrour T (2023) Optimizations of distributed computing processes on apache spark platform. IAENG Int J Comput Sci 50(2):422–433
77. Sundarakumar MR, Mahadevan G, Natchadalingam R, Karthikeyan G, Ashok J, Manoharan JS, Velmurugadass P (2023) A comprehensive study and review of tuning the performance on database scalability in Big Data analytics. J Intell Fuzzy Syst 44(3):5231–5255
78. Biswas N, Mondal KC (2022) Integration of ETL in cloud using spark for streaming data. In: Advanced Techniques for IoT Applications: Proceedings of EAIT 2020. Springer Singapore, pp 172–182
79. Borra P (2024) Comprehensive survey of amazon web services (AWS): techniques, tools, and best practices for cloud solutions
80. Armbrust M, Ghodsi A, Xin R, Zaharia M (2021) Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In: Proceedings of CIDR, vol 8, p 28
81. Kumar A, Mishra A, Kumar A (2024) Build multi-cloud modern distributed data warehouses with Azure and AWS. In: Architecting a modern data warehouse for large enterprises. Apress, Berkeley
82. Simitsis A, Skiadopoulos S, Vassiliadis P (2023) The history, present, and future of ETL technology. In: DOLAP, pp 3–12
83. Ali A, Naeem S, Anam S, Ahmed MM (2023) A state of art survey for Big Data processing and nosql database architecture. Int J Comput Digit Syst 14(1):1–1
84. Patil R, Boit S, Gudivada V, Nandigam J (2023) A survey of text representation and embedding techniques in nlp. IEEE Access 11:36120–36146
85. Silva MC, Eugénio P, Faria D, Pesquita C (2022) Ontologies and knowledge graphs in oncology research. Cancers 14(8):1906
86. Dang NC, Moreno-García MN, De la Prieta F (2020) Sentiment analysis based on deep learning: a comparative study. Electronics 9(3):483
87. Raiaan MAK, Mukta MSH, Fatema K, Fahad NM, Sakib S, Mim MMJ, Azam S (2024) A review on large Language models: architectures, applications, taxonomies, open issues and challenges. IEEE Access 12:26839–26874
88. Mbata A, Sripada Y, Zhong M (2024) A survey of pipeline tools for data engineering. Preprint at arXiv:2406.08335
89. Beretta V (2018) Data veracity assessment: enhancing truth discovery using a priori knowledge. In: Computer Science [cs]. IMT Mines Alès
90. Nambiar A, Mundra D (2022) An overview of data warehouse and data lake in modern enterprise data management. Big Data Cogn Comput 6(4):132
91. Al-amri R, Murugesan RK, Man M, Abdulateef AF, Al-Sharafi MA, Alkahtani AA (2021) A review of machine learning and deep learning techniques for anomaly detection in IoT data. Appl Sci 11(12):5320
92. Lambert SL, Davidson BI, LeMay SA (2023) Survey of emerging blockchain technologies for improving the data integrity and auditability of manufacturing bills of materials in enterprise resource planning. J Emerg Technol Account 20(2):119–134
93. Ding PMR, Wang S Han S, Zhang D (2023) InsightPilot: an LLM-empowered automated data exploration system. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Singapore. Association for Computational Linguistics, pp 346–352

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

## Authors and Affiliations

**Afef Walha[1,2] · Faiza Ghozzi[1,3] · Faiez Gargouri[1,3]**

✉ Afef Walha
afef.walha@gmail.com

Faiza Ghozzi
faiza.ghozzi@isims.usf.tn

Faiez Gargouri
faiez.gargouri@usf.tn

[1] Multimedia, InfoRmation systems and Advanced Computing (MIRACL) Laboratory, University of Sfax, Sfax, Tunisia

[2] Higher Institute of Information Science and Multimedia of Gabes (ISIMG), University of Gabes, Gabes, Tunisia

[3] Higher Institute of Information Science and Multimedia of Sfax (ISIMS), University of Sfax, Sfax, Tunisia