Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi
Bilecik Şeyh Edebali University Journal of Science
e-ISSN: 2458-7575 (https://dergipark.org.tr/tr/pub/bseufbd)

Araştırma Makalesi – Research Article

Geliş Tarihi / Received: 06/12/2024

Kabul Tarihi / Accepted: 28/03/2025

Yayın Tarihi / Published: 08/11/2025

Building An Integrated Database For Turkish Startups: A Systematic And Novel Framework

Türk Startupları İçin Entegre Bir Veritabanı Oluşturma: Sistematik ve Yenilikçi Bir Çerçeve

İsmail Ozan Çelikel[1*], Eda Bahar[2], Günce Keziban Orman[3], Sultan Nezihe Turhan[4]

[1*] Galatasaray University/ Department of Computer Engineering/ Faculty of Engineering and Technology/ İstanbul, Türkiye/ ocelikel@gsu.edu.tr, https://orcid.org/0009-0007-7170-2156

[2] Galatasaray University/ Department of Computer Engineering, Faculty of Engineering and Technology/ İstanbul, Türkiye, ebahar@gsu.edu.tr, https://orcid.org/0009-0000-2243-9266

[3] Galatasaray University/ Department of Computer Engineering/ Faculty of Engineering and Technology/ İstanbul, Türkiye, korman@gsu.edu.tr/ https://orcid.org/0000-0003-0402-8417

[4] Galatasaray University/ Department of Computer Engineering/ Faculty of Engineering and Technology/ İstanbul, Türkiye, sturhan@gsu.edu.tr/ https://orcid.org/0000-0001-9763-0882

Acknowledgements: This study was conducted as part of a research project entitled "Development of Artificial Intelligence Supported, Expertise-Focused, Academic Advisor and Startup Recommendation Portal in R&D Project Management" (Project Number 5240007), which is supported by TÜBİTAK 1505 University's Industry Collaboration Support Program and the GSU Scientific Research Projects under project number FBA-2023-1211

Ethical Statement: It is declared that scientific and ethical principles were followed during the preparation of this study and that all studies used are stated in the bibliography.

Artificial Intelligence Ethical Statement: The author declares that artificial intelligence was not utilized at any stage of the preparation process of this article and accepts full responsibility in this regard.

Conflicts of Interest: The author(s) has no conflict of interest to declare.

Grant Support: The author(s) acknowledge that they received no external funding to support this research.

Teşekkür: Bu çalışma, TÜBİTAK 1505 Üniversitesi Sanayi İşbirliği Destek Programı ve GSU Bilimsel Araştırma Projeleri tarafından FBA-2023-1211 proje numarasıyla desteklenen " Ar-Ge Projeleri Yönetiminde Yapay Zekâ Destekli, Uzmanlık Odaklı, Akademik Danışman ve Yeni Girişimci Firma Öneri Portalı Geliştirilmesi " (Proje Numarası 5240007) başlıklı araştırma projesi kapsamında yürütülmüştür.

Etik Beyan: Bu çalışmanın hazırlanma sürecinde bilimsel ve etik ilkelere uyulduğu ve yararlanılan tüm çalışmaların kaynakçada belirtildiği beyan olunur.

Yapay Zeka Etik Beyanı: Yazar bu makalenin hazırlanma sürecinin hiç bir aşamasında yapay zekadan faydalanılmadığını; bu konuda tüm sorumluluğun kendisine(kendilerine) ait olduğunu beyan etmektedir.

Çıkar Çatışması: Çıkar çatışması beyan edilmemiştir.

Finansman: Bu araştırmayı desteklemek için dış fon kullanılmamıştır.

Building an Integrated Database for Turkish Startups: A Systematic and Novel Framework

ABSTRACT

In recent years, the Turkish startup ecosystem has grown significantly thanks to the increase in government support, diversification of private investments, the spread of startup culture worldwide, and technological developments. Despite the rapidly increasing numbers, there is no up-to-date, comprehensive, and analytically serviceable database for new entrepreneurial firms in different sectors. This study creates an integrated and centralized database for startups in Türkiye by using a hybrid methodology that combines traditional ETL processes with modern data engineering techniques. All data belonging to the companies were obtained from public databases and national techno-hub pools via the web scrape method and stored in a database on MongoDB, a document-based NoSQL database. While data preprocessing provided consistency, integrity, and structural integrity, exploratory data analysis revealed critical insights into the geographical distribution, fields of activity, and workforce metrics of the startup ecosystem in Türkiye. The findings obtained at the end of the study provide very valuable information to stakeholders, including researchers, policymakers, and firms operating in different sectors. The data pipeline construction methodology introduced in the study, characterized by its scalability and adaptability, also serves as a replicable framework for data engineering projects in other fields. Future research can further enhance its analytical capacity by enriching this dataset with financial metrics and sectoral impacts.

Keywords- Turkish Startup Ecosystem, Integrated Database, MongoDB, Exploratory Data Analysis, Web Scraping, DataOps

Highlights

- The Turkish startup ecosystem has grown significantly in recent years, and this study aims to address the lack of an up-to-date and analytically serviceable database by constructing an integrated, centralized startup database for Türkiye.
- A hybrid data engineering methodology was developed, combining traditional ETL processes with modern data engineering techniques.
- Comprehensive data collection was performed using web scraping from public databases, stored and managed on MongoDB.
- Data preprocessing and exploratory analysis ensured consistency and revealed critical information about the geographic distribution, business lines, and workforce metrics of Turkish enterprises.
- The proposed data pipeline offers a scalable, replicable framework that can be applied to similar data-driven projects in other domains, with potential future extensions to include financial and sectoral metrics.

Türk Startupları için Entegre Bir Veritabanı Oluşturma: Sistematik ve Yenilikçi Bir Çerçeve

ÖZ

Son yıllarda, Türkiye'deki startup ekosistemi, devlet desteklerinin artması, özel yatırımların çeşitlenmesi, startup kültürünün dünya çapında yayılması ve teknolojik gelişmeler sayesinde önemli ölçüde büyümüştür. Hızla artan sayılara rağmen, farklı sektörlerdeki yeni girişimci firmalar için güncel, kapsamlı ve analitik olarak kullanılabilir bir veri tabanı bulunmamaktadır. Bu çalışma, geleneksel ETL süreçlerini modern veri mühendisliği teknikleriyle birleştiren hibrit bir metodoloji kullanarak Türkiye'deki startup'lar için entegre ve merkezi bir veri tabanı oluşturmaktadır. Şirketlere ait tüm veriler, web kazıma yöntemi ile kamuya açık veri tabanlarından elde edilmiş ve belge tabanlı bir NoSQL veri tabanı olan MongoDB'de bir veri tabanında depolanmıştır. Veri ön işleme tutarlılık, bütünlük ve yapısal bütünlük sağlarken, keşifsel veri analizi Türkiye'deki startup ekosisteminin coğrafi

dağılımı, faaliyet alanları ve işgücü metrikleri hakkında kritik içgörüler ortaya koymuştur. Çalışmanın sonunda elde edilen bulgular araştırmacılar, politika yapıcılar ve farklı sektörlerde faaliyet gösteren firmalar dahil olmak üzere paydaşlara çok değerli bilgiler sunmaktadır. Ölçeklenebilirliği ve uyarlanabilirliği ile karakterize edilen çalışmada tanıtılan veri hattı oluşturma metodolojisi, diğer alanlardaki veri mühendisliği projeleri için tekrarlanabilir bir çerçeve görevi de görmektedir. Gelecekteki araştırmalar, bu veri setini finansal ölçümler ve sektörel etkilerle zenginleştirerek analitik kapasitesini daha da artırabilir.

Anahtar Kelimeler- Türk Startup Ekosistemi, Bütünleşik Veri tabanı, MongoDB, Keşifsel Veri Analizi, Web Kazıma

Öne Çıkanlar

- Türkiye'deki girişimcilik ekosistemi son yıllarda önemli ölçüde büyümüştür ve bu çalışma, Türkiye'deki startup'lar için entegre ve merkezi bir veri tabanı oluşturarak güncel ve analitik olarak kullanılabilir bir veri tabanının eksikliğini gidermeyi amaçlamaktadır.
- Geleneksel ETL süreçlerini modern veri mühendisliği teknikleriyle birleştiren hibrit bir veri mühendisliği metodolojisi geliştirilmiştir.
- MongoDB üzerinde saklanan ve yönetilen kamu veri tabanlarından web kazıma yöntemi kullanılarak kapsamlı veri toplama gerçekleştirildi.
- Veri ön işleme ve keşifsel veri analizi, tutarlılığı sağladı ve Türk girişimlerinin coğrafi dağılımı, faaliyet alanları ve iş gücü metrikleri hakkında kritik bilgiler ortaya çıkardı.
- Önerilen veri hattı, diğer alanlardaki benzer veri odaklı projelere uygulanabilecek, ölçeklenebilir ve çoğaltılabilir bir çerçeve sunmakta olup, gelecekte finansal ve sektörel metrikleri de içerecek şekilde genişletilme potansiyeline sahiptir.

I. INTRODUCTION

Sartups are fast-growing ventures that offer an innovative product, process, or service that aims to solve a problem in business or daily life. Not every initiative, not every business is a startup. Startups have a high risk of failure, but they can become substantial and influential organizations that shape the sector when they succeed. It is commonly thought that startups are limited to the technology sector due to the impact of digitalization, but many startups play an important role in the economic growth and innovation processes of countries by offering remarkable solutions in many different sectors. The startup ecosystem is particularly dynamic in emerging economies such as Türkiye [1], where government support, private investment, and technological advances combine to create fertile ground for entrepreneurial ventures [2]. The Turkish startup ecosystem also serves global markets by easily adapting to the ever-changing demands of the worldwide economy. However, a review of existing sources reveals that there is no up-to-date, comprehensive, and analytical database of the Turkish startup ecosystem that covers all sectors. This study aims to address this gap by creating an analytically usable, up-to-date database specific to the Turkish startup ecosystem and to enable decision-makers, policymakers, universities, and large companies that want to collaborate with startups to make much more stable decisions [3, 4].

The major motivation behind this study is to contribute to the understanding and development of the Turkish startup ecosystem. The database resulting from the study is a powerful knowledge base for stakeholders in all sectors. The insights gained from this data can be used to set targets to encourage innovation and entrepreneurship in the Turkish business community, both locally and globally.

In the study, a hybrid methodology was applied by combining the data collection, transformation, and storage steps that constitute the classical Extract, Load, and Transform (ETL) processes with recent data engineering techniques. In the first step, we extracted data from publicly available sources of startup data and the most used databases. During this process, we regrettably found out that the vast majority of databases or data stores established to reflect the startup ecosystem in Türkiye contain scattered, inconsistent or limited-scope data. The lack of organized data further enhances the need for a stable dataset and a framework to collect the data. The data sources examined in this study have generally served the purpose of showcasing startups. The data sources examined in this study have generally served the purpose of showcasing initiatives. However, it is important to note that each data source has its own advantages and disadvantages. One of the sources contains exclusively data[1], and the other one restricts[2]. Other open data sources provide inconsistent and outdated information. Additionally, there aren't any data sources that provide comprehensive information regarding the actual fields of activity of startups, the region they are in, their scalability, or their potential for dynamic growth. Throughout our

analysis, the gathered information predominantly consisted of startup names, occasionally accompanied by LinkedIn page URLs or website URLs. However, critical details, such as the current operational status of these companies, employee metrics, and their fields of activity, remained inaccessible. We expanded our search area and scanned all publicly available national, international, and Technopark's' startup databases on the web and collected the data, which we verified with employment platforms, on a data store in order to create a centralized and analytically usable database. We converted the collected data into JSON format to structure it as much as possible by going through the transformation process by addressing issues such as inconsistencies, missing values, redundancies, and/or duplication. To effectively present the dataset to a variety of people and applications and as well as to enhance accessibility, we chose MongoDB, a document-based NoSQL database that is highly compatible with the JSON format. The data was consolidated into a single instance within the MongoDB database. By applying exploratory data analysis and data visualization to the data collected, we also obtained explanatory and predictive insights about the startup ecosystem in Türkiye.

In this study, unstructured data obtained from the web was processed using a pipeline architecture with a modernized ETL process and loaded into a MongoDB NoSQL database. This process was carried out within the framework of DataOps principles, aiming to automate the data flow, optimize the data processing and transfer steps and maintain data quality. Thanks to the pipeline architecture, data processing flexibility was increased and delays in data flow were minimized. While the flexibility and scalability provided by using NoSQL solutions such as MongoDB comply with the agility principle of DataOps, the traceability and quality control of data processing steps throughout the process supported continuous improvement and reliability in data processes. This study contributes both academically and technologically as an innovative application of DataOps in data management processes. In addition, from a data engineering point of view, the technologies used in the study and the scalable methodology developed can be used in projects requiring data engineering in different fields. From this point of view, the paper presents a replicable framework to properly manage the complexity of modern data ecosystems, both by effectively using a document-based NoSQL database and by modernizing traditional ETL processes.

The remainder of the paper is organized as follows: Section 2 explains the current methods in the literature regarding the current studies on startup data and related technologies, Section 3 discusses the data extraction, transformation, and storage processes in depth, describing web scraping techniques and ETL, storage, and data processing methods. Section 4 presents the main findings from the descriptive analysis of the built system's components, infrastructure architecture, and data and sheds light on the trends, challenges, and opportunities in the Turkish startup ecosystem. Section 5 presents the implications of the findings and recommendations for future research and practical applications. This study will provide a basis for all possible strategic planning by addressing the current shortcomings in data availability and reliability in unlocking the potential of Türkiye's startup ecosystem, and the methodology used in the study will serve as a guide for similar studies.

II. RELATED STUDIES

Despite the lack of consensus on its definition, a startup can be defined as a nascent entrepreneurial venture characterized by a team which shares ownership and decisions rights. Knight et al., define start-up as a team in which team members have a financial interest, the team possesses decision-making authority and agency, and the team is a social entity with distinct boundaries [5]. The majority of start-ups, which have a great importance for national economies with their job creation, innovation, industry transformation, regional economic growth and wealth creation features, fail in the very early stages of their participation in the labor market. Various statistics show that start-up failure rates are around 90 per cent [6].

As noted in Mandel's paper, a healthy startup ecosystem is essential for economic growth and job creation and requires policies that support access to capital, talent, markets, and innovative regulatory and fiscal policies. The primary objective of governments should be to encourage the formation of new firms that can grow rapidly across a range of industries and regions [7].

As Basole et al. point out in their paper [8], data is vital in building a healthy startup ecosystem, as it is in all areas of business. Data provides valuable information for entrepreneurs, investors, mentors and policy makers and supports the development of the ecosystem. A database with robust, reliable and continuously updated data on startups is always necessary for stakeholders in this field. There are continuously updated databases around the world with a particular focus on startups [9]. For example, Ticu, in his master study, analyzed the startup ecosystem in Austria and conducted various analyses on the data collected. Jáki et al. conducted a study on Characteristics and challenges of the Hungarian startup ecosystem [10]. In addition, there are websites that serve startup data especially to angel investors and companies that want to cooperate/partner. Many of these websites also provide this service for a fee.

In Türkiye, data on startups are generally shared through networks, such as Technopark websites, TÜBİTAK Bigg, LinkedIn. In addition, Türkiye Technohub Initiative [11], which is supported by the Digital Transformation Office of the Presidency of the Republic of Türkiye, only has data on startups operating in the field of technology. The data served by other platforms is generally irregular, low quality and outdated.

Considering the high failure rate of startups, it is a great necessity to create an up-to-date, accurate and constantly renewed database.

In the study initiated to create an up-to-date and centralized database on startups participating in the labor market in every sector, the first step was to collect data from reliable sources. This process, which is estimated to take a very long time with traditional methods, has been accelerated with the web scraping data collection method. Massimino defines web scraping as the systematic collection of publicly available data [12]. However, as Mancosu and Vegetti pointed out in their study, many websites include various security measures to restrict bot traffic [13]. However, Krotov and Silva state that web scraping can be applied within an ethical and legal framework within the scope of scientific research in the public interest [14].

Web scraping goes beyond traditional data collection methods by providing access to large volumes of data. It is a powerful method for obtaining data from websites and can provide access to large volumes of data far more than the amount that can be collected by traditional methods such as official statistics or surveys [15]. Thanks to automation tools such as Selenium, crawling processes that simulate user interactions can be realized [16]. In literature, this method has a wide range of applications from social sciences to medicine, from marketing to financial research [17]. For example, Luscombe et al, in their study, described how web scraping techniques should be used for social science research using algorithmic thinking in the public interest [18]. Goulas and Karamitros discussed methods of data collection with web scraping for medical and surgical research [19]. In their research, Rodrigues and Polepally have focused on creating a financial database for educational and research purposes using web scraping techniques. They demonstrate how to collect and organize publicly available financial data from various websites, which can be used for research, education, and also other purposes. In addition to Python Beautiful Soup and Selenium libraries, they also used Scrapy, lxml and Pandas [20]. Styawati et al. focused on the use of web scraping in a very different area. They investigated how web scraping can be used to collect data from multiple freelance job websites to create a streamlined job search experience for users over the web. In the study, they used a special combination of BeautifulSoup and Selenium libraries in Python for web scraping, which made it easier to scrape data from the general structure of freelance job websites in their country [21]. Barba et al. address a very different application of web scraping in their study. The study explores the integration of AI models and web scraping techniques in businesses and highlights four potential research areas: Machine Learning for sentiment analysis, AI and Natural Language Processing (NLP) integration, Data intelligence and optimization, and NLP and Deep Learning (DL) integration. They provide a theoretical and practical overview of emerging research directions and encourage managers to adopt advanced AI-based models to enhance the value of web data obtained through scraping [22]. As can be seen, web scraping is a versatile tool that supports a wide range of disciplines by providing access to valuable data that is difficult to obtain with traditional methods. The applications of this method, of which we have listed a few examples above, in many different fields make it an important technique in the modern data-driven world.

The data collected by the web scrape method is not expected to have a singular format. As Sudarajat et al. stated in their study, the integration of data of different structures collected from different sources brings difficulties such as data inconsistency, complexity, reliability and technical difficulties [16]. As Zhou et al. stated in their article, websites often exhibit different structures and formats, making data collection and organization difficult [24]. Furthermore, as Spaniol et al. state, the reliability of web data can vary significantly, and the open and anonymous nature of the web has a negative impact on data quality [25]. Finally, as Vording points out in his paper, a significant proportion of web data is unstructured and requires intensive transformation for analysis [26]. In order to overcome all these challenges, it is essential to create a very powerful data integration architecture.

As Gandhi et al. point out in their work, the cycle of collecting, processing and preparing data for use is long, but there are modern solutions that simplify and improve this process [27]. These solutions include the traditional ETL process [28], which is an integral part of BI projects, the ELT process [29], which emerged with big data analytics, and automated data pipelines [30], which were created specifically for processing real-time data. Undoubtedly, each of these methods can be used individually or all together in the collection and transformation of web scraped data.

While traditional ETL processes provide the cleaning and transformation of data in a specific format, data pipelines processes allow large-scale data to be moved automatically [31]. The focus of the ETL process is to clean and transform the data in accordance with the target data schema. The focus of data pipelines is the automation of data flow in a scalable way. A traditional ETL process is mostly responsible for taking data from dedicated sources, transforming it and moving it to the target database. Even under these conditions, the ETL process usually consumes more than 70 per cent of the available resources for projects. As Nwokeyi and Matovu state in their article, it is very time-consuming and challenging to move high volumes of data in many different formats and without clean content according to the traditional ETL process [32]. On the other hand, web scraping does not only mean collecting structured or semi-structured data produced by APIs. In the web scraping process, data collected directly from the raw HTML of the site is also obtained. As Walha et al. point out in their work, it is a natural part of traditional ETL to integrate data from different data sources and transform them into the format imposed by the target system. In contrast, data pipelines usually focus on moving large amounts of raw data quickly, so deep

integration of data is not usually the focus of this process [31]. On the other hand, as Bhatlawande et al. emphasize in their work, traditional ETL processes are starting to experience serious performance and scalability issues with the increase in data, while data pipelines can manage large volumes of data by leveraging technologies such as Docker, Kubernetes and distributed systems [33]. It is necessary to create a scalable architecture that can easily adapt to the schema changes of web scraped data, provide smooth data flow, and integrate data from different sources. This architecture, which Hafyani et al. call ETL data pipeline [34], transforms ETL processes in accordance with the structure of big data, optimizes workflows and resource usage, and makes the integration process more efficient. This architecture has the capabilities to overcome the challenges of data integration over the web by providing efficient data extraction, transformation and loading features. While supporting the integration of data collected from different sources from the web, they also offer automation, and optimization features to increase performance and scalability.

The superiority of ETL processes over data pipelines is particularly emphasized in situations that require tight control over data integration, transformation and data quality. However, in modern data projects that require real-time data processing, large data sources and flexibility, data pipelines can be more advantageous. These features are also reflected in the storage of the collected data [35]. In their study, Diouf et al. defined the ETL process as the process of collecting data in different formats from multiple sources and transferring them to a single central data warehouse and emphasized the importance of the storage process in integrating and centrally managing the data [36]. This means consolidating the data, which significantly improves the quality of the data [37]. Data pipelines are generally more flexible than ETL and can handle different types of data sources. Therefore, data storage methods also vary to reflect this flexibility. Since pipelines are usually built for the management of large volumes of data with constantly changing/increasing volumes, storage systems must also be flexible and scalable. At the same time, given the real-time data flow, it is of great importance to ensure both the security of the data in the storage process and the ability to quickly restore it from backup when necessary [38]. In short, the data storage strategy in both ETL processes and data pipelines is an important factor that directly affects the performance of the system, data quality and security of the data. In traditional ETL processes, the preferred method for data storage is relational databases. The data processed and transformed in the process is loaded and served from a data warehouse whose schema is rigidly pre-designed. In data pipeline architectures where flexibility and scalability are at the forefront, NoSQL databases are preferred on cloud infrastructure in order to optimize the use of resources [39]. When deciding whether to load web scraped data into a relational database or a NoSQL database, the choice largely depends on the nature of the data and the specific requirements of the application. As Khan et al. point out in their paper, NoSQL databases are very good in terms of performance for write-heavy operations, which is useful for applications that continuously generate large amounts of data. Relational databases, on the other hand, can handle particularly complex queries efficiently and often exhibit high performance in applications that require complex joins and operations [40]. For large volumes of unstructured data that require high scalability and flexibility, NoSQL databases are more suitable, as stated in the study conducted by Ali et al. On the other hand, if the data is structured and the application requires strict data integrity and complex querying, a relational database may be a better choice [41]. The decision should be based on the specific needs of the application, considering factors such as data structure, scalability and performance requirements. As Ambre et al. state in their work, MongoDB fulfils all the above-mentioned constraints, while web scrape is a scalable NoSQL database where data can be easily stored [42]. According to Rathor et al, MongoDB's indexing capabilities as relational databases enable efficient querying of large datasets, which is crucial for handling extensive web crawling results. The database's horizontal scalability through sharding allows for improved performance as data volume increases. It also offers better query performance compared to traditional relational databases when dealing with large amounts of data. Its ability to handle high-speed data ingestion makes it suitable for web scraping applications [43].

DataOps applies DevOps principles to data management processes, making data flow faster, more reliable and collaborative [44]. Bergh et al. state that the DataOps methodology helps big data projects to be executed more effectively by improving data quality. In the aforementioned book, the authors explain that DataOps methodology accelerates and automates data processes, makes data available quickly, reduces data errors through automation and quality control processes, provides more consistent data, and quickly adapts data infrastructures [45].

In the study developed within the scope of the article; after scraping unstructured data from the web, it automates the data flow, data processing and data transfer steps by uploading the data to MongoDB through a modernized ETL process. In this process, high quality data will be obtained by being cleaned and integrated into a single center and converted into a single format. In addition, since a data pipeline that will work end-to-end is established and the traceability of the data is increased, the principles of DataOps methodology are applied.

III. METHOD

In this project, we aim to construct a comprehensive dataset on startups in Turkey by systematically collecting, storing, and maintaining the data within a sustainable framework. By implementing a unified architectural approach, we ensure that both existing and newly acquired data are seamlessly integrated, enabling the dataset to remain dynamically updated. Furthermore, this framework will facilitate future research and

analytical operations on the dataset. Following the design of the data collection, Data-Driven Analysis (DDA) techniques will be applied to extract meaningful insights from data, contributing to a deeper understanding of the startup ecosystem.

This chapter is structured as follows: Section A outlines the data extraction strategies, while Section B details the implemented ETL processes. Next, Section C examines potential data storage strategies, and finally, Section D defines and explains the applied Data-Driven Analysis (DDA) methods.

A. Data Extraction

Data scraping, also known as web scraping, is the process of automatically obtaining information from websites or online sources [15]. This method is widely used in various fields, such as data analysis, research, marketing, and software development. In Data Scraping, the primary goal is to collect large volumes of data from the web autonomously. Instead of manually copying and pasting data, scraping automates the process, allowing for the efficient extraction of data from multiple pages or sources [46]. This method is useful for allowing data scientists and other researchers to gather data from web sources [47] but is also used in different fields such as marketing [46], e-commerce [48], tourism [49] and other startup research [50] for building datasets and also providing insights to decision makers [46].

A scraper is a tool, script, or program designed to extract data from websites. To access the content of a web page, the scraper must first communicate with the web server hosting that page. This communication is initiated by sending a request [15]. We realize this process using HTTP (Hypertext Transfer Protocol) which serves as the foundation for data communication on the World Wide Web. HTTP establishes a set of rules for transferring files, such as text, images, video, and any other data formats, from a web server to your browser. These files are generated with HyperText Markup Language (HTML), the standard language used to create a website. It provides the basis for creating web pages and determining the layout, formatting, and basic appearance of a web page. It is mostly used with CSS (Cascading Style Sheets) to style the website.

The first step in the process of extracting data from websites is to identify the pages that contain the relevant data. This process involves analyzing the structure of the website and determining in which area the relevant data resides since each HTML page has a unique Document Object Model (DOM). The DOM is a hierarchical representation of HTML documents and defines all elements on the page as nodes. The browser generates the DOM from the HTML document it reads, and as a result, the content of the page is presented as a tree-like data model. This makes it possible to access and modify all the components of the web page, such as headings, paragraphs, images, links, input fields, etc. through programming.

In the data extraction process, the targeted data is located in nodes or clusters of nodes in the DOM tree. The correct identification of each node ensures that the right data is accessed. Web scrapers use tools such as CSS selectors or XPath to navigate the DOM and retrieve HTML elements [47].

Once the target pages have been identified, an HTTP request needs to be sent to access the web page. This is usually done using GET to retrieve data or POST to send data. The GET request asks the server to return the content of the specified page, which is retrieved as the DOM. At this point, the structure of the DOM is the basis for analyzing the loaded page and selecting specific data. In addition, the DOM allows not only to programmatically change and analyze the structure of the page, but also to perform interactive operations on the page. This is especially important for handling dynamically generated content in JavaScript, as it makes it possible to trigger scripts running in the browser to extract the data correctly.

Currently, there are many different libraries and programming languages available to perform this process. Among these, “Beautiful Soup” and “Puppeteer” libraries are particularly noteworthy [51]. Beautiful Soup [52], a Python library, is frequently used for parsing HTML and XML documents. Retrieving web page content becomes extremely easy, especially when used in conjunction with the “requests” library. It transforms the web page into a parse tree, making both data extraction and navigation and search over HTML components more efficient and easier. Puppeteer [53], a JavaScript library, provides a high-performance API for controlling Chrome or Chromium browsers. It automates all web-based tasks by making it easy to manage actions such as clicking buttons, filling out forms, and navigating between pages. This makes it easy to scrape all the data resulting from an action performed on a page.

In the data extraction phase, scraper collects content such as text, URLs, images, and tables from within HTML tags. It is evident that not all websites are constructed using the same HTML code. For this reason, the scraper's code should be customized according to the website it accesses. Puppeteer is also powerful in analyzing the HTML code of the website it accesses and finding out which tags contain the desired information. Using two powerful libraries together can make data scraping quite easy and fast.

B. ETL Process

Data storage is comprised of many software modules, which are responsible for populating the data storage with fresh data. The process begins with extracting data from an appropriate data source, followed by transforming it to meet the requirements of the target storage solution, and concludes with loading the data into the storage. These software modules are known as Extract-Transform-Load (ETL) activities. Each step of ETL processes can comprise different software modules and solutions, such as scripts written in a programming language, built-in functions of a storage solution, calls to an external library, and so on. [51] These various ETL modules aim to prepare the extracted data for loading to the data storage and define all the steps the data goes through between the data source and the data storage, as seen in Figure 1. During this process, the extracted data is transformed to comply with the needs of the loading stage. In the transformation stage, the data is transformed into a common data model, the errors and null values are removed, data values are standardized, data is integrated into a single consistent data set, duplicates are removed, and finally, sorted. With these steps done, the data is ready to be loaded into the data storage. These stages are generally comprised of different software modules and, for that reason, are generally implemented specifically for the needs of the workflow at hand. [54]

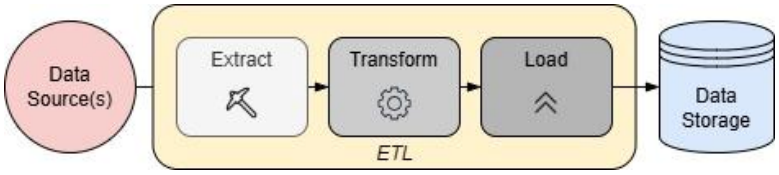


Figure 1. High-level design of the ETL process.

In this study, the *extract* stage involves gathering data from web scrapers and other sources. During the *transform* stage, the data is prepared and converted into a format compatible with the selected storage solution. Finally, the ETL process concludes with the *load* stage, where the processed data is inserted into the storage system.

Table 1. Summary of data storage solutions.

Data Storage Solutions	Advantages	Disadvantages
Relational Databases (RDBMS)	Strong data integrity and consistency (ACID compliance).	Limited scalability, particularly horizontal scaling.
	Well-suited for complex queries and structured data.	Rigid schema design can be inflexible.
	Mature and widely supported.	Less efficient for unstructured data.
NoSQL Databases	Flexible schema design, ideal for unstructured or semi structured data.	Weaker consistency guarantees (CAP theorem).
	Scales horizontally easily.	Less mature tooling and standardization.
	High performance for read/write operations.	Complex queries can be less efficient.
Object Storage	Excellent for storing large volumes of unstructured data (e.g., images, videos).	Not optimized for complex queries.
	Cost effective and scalable.	Higher latency compared to block or file storage.
	Accessible via APIs.	Weaker consistency models and data retrieval.
Data Warehouses	Optimized for analytics and reporting on structured data.	Expensive to maintain and scale.
	Supports complex queries across large datasets.	Not ideal for real-time data processing.
	Centralized data repository.	Rigid schema design and data integration complexity.
Data Lakes	Capable of storing structured, semi structured, and unstructured data.	Data management and governance challenges (e.g., data quality).
	Highly scalable and cost effective.	Slower performance for querying.
	Ideal for big data and ML.	Security and access control issues.

C. Data Storage

Undoubtedly, the most crucial phase of a data-driven study is to develop a meticulously defined data storage strategy. It is of great importance to determine the most appropriate data storage strategy to ensure integrity,

consistency, accessibility, and security during data use. The first step in determining a data storage strategy is to understand and interpret the data. It is necessary to define and determine the structure and characteristics of the data, the types of applications to be implemented with this data, and the purposes of using the data from the very beginning [55].

The second crucial stage of creating a data storage strategy is to determine the data storage solutions and data storage infrastructure solutions to be used. Data storage solutions cover a range of options suitable for various purposes [56]. Table 1 shows the advantages and disadvantages of data storage solutions.

Among the data storage methods listed in Table 1, relational databases and NoSQL databases stand out as the two most obvious solutions. Both solutions have their own advantages and disadvantages. Although relational databases are designed to store data, their primary function is to provide up-to-date, consistent, secure, and integrated data for use in transactional systems. To achieve this, all relational databases must comply with a set of rules known as ACID rules. As a result, data in relational databases must conform to a predefined and rigid schema from the moment they are created. These rules facilitate the production of secure and accurate information in relational databases, but they also cause data flexibility to disappear completely. [40] Data warehouses, designed to support business intelligence and data analytics systems, have a much more flexible data model compared to transactional relational databases. However, they still need to conform to a database schema. This is because all traditional data warehouses are built on relational data models and relational database management system software. [57] However, in recent years, with the emergence of the concept and principles of Big Data, the need for these rigid schemas has been replaced by a much more flexible schema, or even a demand for a schemeless data structure. A data lake is a centralized storage solution that allows the collection, processing, and analysis of significant amounts of raw data in various formats [58]. In this storage solution, data in various structures can be stored together, including structured (e.g. data in databases), semi-structured (e.g. XML, JSON), and unstructured (e.g. text, images, audio). [58] Unlike data warehouses, which are designed to host only structured data per a predefined schema to perform analyses on a specific subject, a data lake solution allows the storage of different data types and formats in a way that can be modeled according to emerging needs without the need for a schema. There are various tools available for creating a data lake, either on-premises or in the cloud. The basic common point among these tools is that they are distributed file systems [57]. However, in recent years, research on the potential of various data storage tools for the creation of data lakes and the management of large-scale data has increased. NoSQL databases represent a data storage solution supported by flexible and customizable data models. They are designed to provide superior accessibility, scalability, and performance compared to traditional relational databases. These databases operate on distributed architectures, providing horizontal scalability while also meeting critical needs such as data flexibility and transaction speed. Optimized specifically for big data analytics and real-time data processing, NoSQL databases support high availability, but at the expense of weaker consistency. In addition to being more suitable for processing large volumes of data, they are also more capable of processing data representing various types and formats. [40] As explained in [41], NoSQL databases are classified into four main categories according to the data model that forms the basis of their architecture:

1) *Column-store NoSQL databases*

2) *Document-based NoSQL databases*

3) *Key-value NoSQL databases*

4) *Graph databases*

NoSQL databases are also suitable for processing various types of data in data lakes due to their advanced scalability and strong performance features [59]. Several studies in the field propose mechanisms for transferring data from data lakes to NoSQL data warehouses [60]

While deciding about a data storage solution, it is necessary to determine simultaneously the data storage infrastructure as well. In this process, it is necessary to consider factors such as data volume, required storage capacity, query performance tolerance, and scalability [61]. This decision ultimately determines whether a cloud solution or an on-premises system is required as the infrastructure architecture.

Three other factors that significantly affect the determination of an optimal data storage strategy are the data model, data pipeline, and backup operations. The data model determines how the data will be organized, how functional associations and dependencies will be built, and the impact of these structures on system performance. Additionally, it determines the methods of storing and accessing the data. NoSQL databases provide flexibility through different data models and configurations, including document-oriented, column-oriented, key-value, and

graph databases. In this way, dynamic data structures are processed much faster, and scalable architecture is provided. Accordingly, the model structure of the data and application requirements are the primary factors that inform the selection of a suitable storage solution. In addition, the degree of consistency demanded by the data model is extremely important in the selection of the storage methodology, which affects data replication and distribution strategies [62]. The data line, on the other hand, occupies an important place in the data storage process due to the movement of data. As a result, the fluidity and performance of the data line are directly related to the scalability and integration ability of the storage method. In addition, backup operations are extremely important to prevent data loss and ensure data security. Therefore, it is essential that the selected data storage method allows backup and recovery operations to be carried out effectively and facilitates these processes. Failure to consider these factors together can lead to significant problems in terms of data management and sustainability in the long term.

In this study, MongoDB was selected as the data storage solution due to its flexibility and scalability, which align with the requirements of processing various data types in large volumes. MongoDB's document-based NoSQL structure [63] allows for the storage of semi-structured data like JSON, making it highly compatible with the dynamic and evolving data model used in this study. Additionally, its ability to scale horizontally, combined with its strong performance for CRUD operations [64], made it an ideal choice for handling the diverse data extracted from web scrapers and other sources.

D. Data Preprocessing and Descriptive Analysis

The ability to produce accurate results in any kind of analysis, from classical methods such as basic statistical analysis, time series analysis, and regression models to more advanced approaches using machine learning and artificial intelligence algorithms or complex analyses such as data analytics, clustering, and forecasting performed on large datasets, depends on the quality and reliability of the data used.

1) *Data Preprocessing:* Data preprocessing is the process of cleaning, transforming, and structuring raw data [65]. It is especially crucial while working with real-world datasets [66]. Preprocessing performance is a critical step that directly affects the accuracy and reliability of the insights to be obtained as a result of the analysis and, therefore, the success of the type of analysis performed [67].

The Data Preprocessing step, which aims to clean, transform, and prepare raw data, thus transforming it into a format of quality and usability for subsequent analysis, consists of the following tasks:

- Data Cleaning: This task involves handling missing data, correcting errors, and removing noise from the data. Techniques such as inputting missing values based on available data or removing rows or columns with excessive missing values are used in this step. Outliers are also detected and corrected either by deletion or transformation [68].

- Data Transformation: This process involves scaling, normalizing or encoding the data to improve its compatibility with the algorithms and models to be used. Data normalization and standardization optimize the effectiveness of models, methods or algorithms used in data analytics. These methods require transforming data into a uniform scale or distribution, thus ensuring that all features contribute equally to the analysis and preventing larger-scale variables from dominating the results. For example, in the context of machine learning applications, features can be normalized to ensure that they exhibit a similar range. Furthermore, categorical variables can be encoded using techniques such as one hot encoding or labelling encoding to convert them into numerical representations that can be understood by algorithms. Continuous variables can also be discretized, when necessary, into categorical variables with different intervals [68].

- Data Integration and Reduction: Often, data originates from multiple datasets or databases and requires integration to create a unified dataset. This may require resolving schema mismatches and combining features from different datasets. Furthermore, data reduction techniques such as principal component analysis (PCA) can be used to reduce the size of the dataset while retaining its key features [69].

- Feature Engineering: In many cases, especially when working with real-world data, the raw features contained in the dataset can negatively affect the performance of the model. Feature Engineering involves creating new features based on the raw features available and reshaping existing ones in a different format to improve model performance. This step has a significant impact on the performance of models [70].

2) *Data Descriptive Analysis (DDA):* Data Data Descriptive Analysis (DDA) is the process of summarizing and visualizing key features of a dataset, usually as a first step before more complex analysis. It provides insight into the distribution of the data, relationships between variables, and patterns that can inform subsequent modeling decisions [71]. The basic techniques are as follows:

- *Summary Statistics:* Measurements used to obtain a basic understanding of the central tendency and spread of the data. The most used measures in summary statistics are mean, median, and mode, which are used to understand central tendency. The mean is an overall indicator of the Center of the data set. The median represents the middle value in sorted data and is used to understand the Center point in skewed distributions. The mode represents the most frequent value(s). Especially for categorical variables, frequency counts and mode are commonly used. Mean, median, and mode are also commonly used in data preprocessing to fill in missing data. Apart from these measures, we can talk about measures of dispersion, range, variance, and standard deviation. The range reveals the difference between the highest and lowest values. Variance gives an idea of how far the values are from the mean. Standard deviation determines the extent to which observations in the data set deviate from the mean, revealing whether they exhibit a broad distribution or are clustered in a narrower range. It also helps to identify outliers in the data set [72].

- *Distributional Analysis:* Understanding the distribution of each attribute is important for selecting appropriate models and methods. Measures of skewness and kurtosis are often calculated to assess the symmetry and skewness of distributions. Shape measures such as skewness and kurtosis describe the shape of the distribution. Skewness measures the asymmetry of the data distribution and indicates whether the data is skewed to the left or right. Kurtosis, on the other hand, indicates whether the data has thicker or lighter tails than a normal distribution. Identifying non-normal distributions can encourage the application of transformations or non-parametric methods in further analysis [73, 74].

- *Correlation Analysis:* In DDA, correlation analysis focuses on the relationship between variables, often measuring the strength and direction of relationships between numerical variables by constructing correlation matrices. For example, Pearson correlation coefficients reveal linear relationships. This gives insight into how changes in one variable can be associated with changes in another variable. Spearman's rank correlation, on the other hand, reveals monotonic relationships in non-linear data. In this way, meaningful results can be obtained even when the relationship between variables is non-linear or the strength and direction of the rank relationship between data can be more clearly determined [75].

- *Data Visualization:* Visualizations such as histograms, box plots, and scatter plots provide a graphical representation of data distributions and relationships between variables. These charts help identify trends, clusters, and outliers that may not be immediately apparent from summary statistics alone [76].

IV. EXPERIMENTS AND RESULTS

As mentioned in the previous sections, the primary objective of this study is to create an analytically usable database for startups in Türkiye. The extracted data must be transformed into a structural format so that it can be used for further analytical studies. In the previous chapter, we have defined the methodology for creating the said database. In this chapter, the results of the methods will be examined, which includes how the data sources are selected, how the data is extracted from these sources, transformed and loaded into data storage, and finally, how it is analyzed.

A. Data Sources

The first and vital step towards creating this database is identifying the sources from which the relevant data will be collected. Even though Türkiye's first and largest local employment platform, Kariyer.Net, is one of the project stakeholders, when we examined the data in the databases owned by the company, we observed that it was insufficient. Since the company mostly serves organizations above a certain scale, the number of startups registered in its database was very small. In the second stage, we identified the websites where startup data is generally made publicly available. Since our main goal was to create a database specific to Türkiye, we first browsed the websites of Technoparks, currently operating in different cities in Türkiye. As a result of this scanning process, the websites of TÜBİTAK BIGG, which operates under the Scientific and Technological Research Council of Türkiye, and ITU Çekirdek, which operates at Istanbul Technical University, came to the fore. However, the examination of these websites and the data they provide revealed that the information they contain is inadequate and incomplete, failing to meet our expectations. The Turkish Technohub platform [77], provided by the Digital Office of the Presidency of the Republic of Türkiye, although it offers incredibly high quality and clean data, was limited in terms of sectoral aspects as it only contains data on startups working in the “Technology” sector. Nevertheless, we still use this website as a reference to verify the data we collected from our other sources. All these constraints led us to search different global websites and scrape data from them. As a result of this search process, we found that StartupWatch [78] and StartupMarket [79] websites contain sector-independent, up-to-date, clean, complete, and comprehensive data about the startups in Türkiye.

In order to scrape the selected sites, the websites are thoroughly analysed. This is a crucial step in the preparation of the web scrapers, as they must be designed in accordance with the structure of the website. The

HTML codes of each website are analysed to detect which sections of the code contain the required information. The general structure of a web scraper can be defined as follows:

1) Finding the URL: The first step in scraping the websites is to find the URLs for startups. In our case, the websites have pages listing all the startups. These pages are ordered in ascending order, so the scraper can browse through these pages by modifying the URL. By deciding on the URLs, the HTML response of the URL can be acquired by using the "requests" library for Python [80], which is a library that specializes in handling HTTP requests.

2) Parsing the HTML code: While browsing these pages, the HTML code of the website must be parsed in a way to detect the required sections. This is done by following the hierarchical structure of the HTML body. To this end, a Python library for web scraping "BeautifulSoup" is used, which is primarily used for modifying, searching, and navigating through a parse tree.

3) Extracting the data: After detecting the necessary sections of the HTML code, the data is extracted from these sections (such as startup name, location, field of operation etc.). The extracted data is loaded into a JSON object, which is then sent to the data controller unit for the remaining steps.

B. ETL Process

Following data extraction, the data must be prepared for the loading stage. In the remaining ETL steps for startup data, the goal is to process the JSON data collected by parsers, ensuring compatibility with MongoDB and improving data quality. These steps can be summarized as follows: (1) Parsing the data, (2) Transforming the data, and (3) Loading the data. All stages are implemented using Python and illustrated in Figure 2.

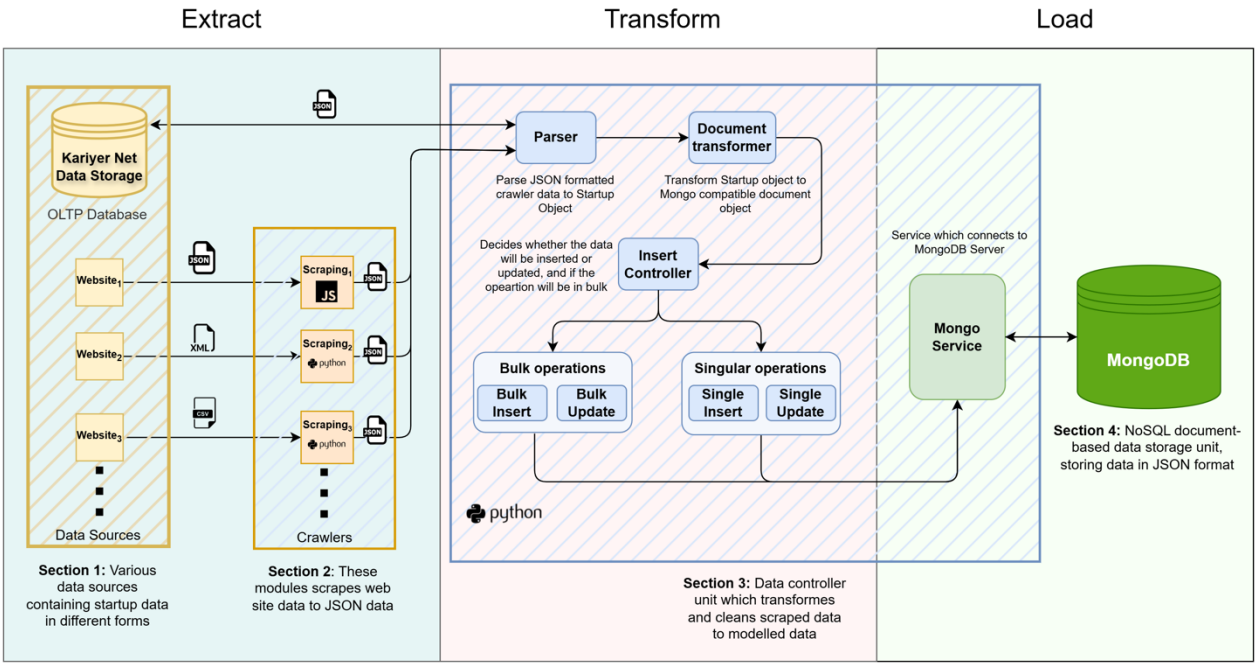


Figure 2. Diagram of the ETL process.

1) *Parsing the Data:* In this stage, data from web scrapers and other sources, such as the Kariyer.Net Data Storage (Section 1 of Figure 2), is parsed. A Startup class is created to store the data in JSON format. This object-oriented approach enables efficient manipulation and formatting of the data. To ensure consistency, a set of rules is applied during parsing. These include capitalizing the initials of country names, standardizing company addresses, correcting spelling errors, and defining a unique identifier for each startup. This phase lays the groundwork for transforming the parsed data into a structured format compatible with MongoDB.

2) *Transforming the data:* The transformation stage involves preparing MongoDB-compatible documents (as shown in Section 3 of Figure 2). MongoDB stores data in BSON (Binary JSON) format with a formatting similar to JSON, a flexible data structure. In this way, complex data hierarchies can be easily modeled, and database queries can be made more efficient. To comply with this need, the existing Startup class is encapsulated with a Startup document class, essentially implementing a DTO (Data transfer object) pattern [81]. This pattern minimizes the method calls of the main Startup object. This approach not only enhances efficiency but also supports the flexible modelling of complex data hierarchies.

3) Loading the data: At this stage, all the data collected from scrapers and other sources was parsed into the Startup class which is implemented in Python language and converted into JSON format to be compatible with MongoDB. Transferring the data to MongoDB was easily realized through the Pymongo library. Pymongo is the official MongoDB driver for Python and supports CRUD (Create, Read, Update, Delete) operations and other MongoDB operations [73]. In this study, Pymongo is used to perform both update and insert operations. These operations were performed in both singular and bulk mode. Bulk operations provide an efficient solution when a large amount of data needs to be added or updated to the database, and update operations are preferred over insert operations when the document already exists. This method also provides an efficient strategy to avoid data duplication in the database.

ETL activities are designed to align with data modeling requirements. The primary goal during modeling is to ensure data consistency while accommodating the flexible nature of the extracted data. Data collected from various sources, such as scrapers and OLTP systems, is automatically extracted in formats that may vary significantly. This variability necessitates a flexible modeling approach. Data points can contain empty, numerical, textual values or even arrays and objects. To handle this diversity effectively, semi-structured data formats are preferred.

Among the available formats, such as XML, Parquet, and BSON, JSON is chosen for this study due to its compatibility with MongoDB and widespread use in web applications. MongoDB stores documents in BSON (Binary JSON) but allows insertions in JSON format [82]. Additionally, JSON is language-independent and a standard for data interchange in web applications, making it an ideal choice.

Another crucial aspect of data modeling is ensuring data consistency across multiple sources, which may include overlapping information about the same startup. This overlap can lead to duplicate data during the loading stage. To address this, a merging strategy is employed in the transformation stage. Each data point receives a unique identifier generated from the startup's name. When a new data point is inserted, its identifier is queried against the storage. If a match is found, the existing data is cross-checked with the incoming data. Any new or differing fields are updated, ensuring no duplicate entries are introduced into the database.

Normalization is also applied to enhance data quality. After duplicates are eliminated using the method above, inconsistencies in company names are resolved by standardizing capitalization and removing special characters. Fields containing dates, such as the establishment date, are standardized to a uniform format. Similarly, categorical fields, such as the startup's field of operation, are normalized to ensure consistency across all records. These transformation tasks—normalization, deduplication, identifier assignment, and formatting into JSON—are managed by the data controller module.

With these processes complete, the extraction and transformation steps are finalized, and the data is ready for loading into the storage solution. Loading can be performed either in bulk or individually, as depicted in Figure 2. Insert controllers interact with the Mongo Service, which completes the loading stage.

C. Data Storage

When a storage option is selected by the criteria defined in the Data Storage section of the Methodology chapter, the qualities of the data must be taken into consideration. Additionally, the capabilities provided by the storage solution must be accounted for. On this premise, the following steps are taken to decide on a storage strategy:

1) Interpret the data: In this study, the data we collected from various sources (e.g., from OLTP database of Kariyer.Net and/or via web scrapers) was initially either structured or semi-structured (in JSON format). The fields of the data after the transformation stage are described in Table 2. These fields can contain null values as the data source may not contain the field. Additionally, the data contains information for 4319 startups, and this volume can change with further iterations. This variety in the data and the possible increase in the volume of the data shares similarities with the qualities of Big Data characteristics. An example startup instance from MongoDB is illustrated in Figure 3.

```
_id: "cybeerly"
research_fields : ""
description : "Cybeerly is a platform providing a virtual classroom with innovative t…"
website : "cybeerly.com"
corporatization : "E"
linkedin_address : "https://www.linkedin.com/in/rufat-gulmalizade-819a07214/"
establishment_date : "1/1/2021"
location_country : "Türkiye"
logo : "https://startupmarket.co/cache/360x180/assets/img/no-image.jpg"
company_name : "Cybeerly"
employee_count : "4-10"
location_city : "İstanbul"
field_of_operation : "SaaS & PaaS, Eğitim"
```


Figure 3. An example startup instance from MongoDB

2) *Storage Type:* After the transformation stage, the final format of the data is the JSON format, which allows flexibility in the data representation. NoSQL databases have advantages when processing JSON documents as they are better suited for semi-structured data formats.

3) *Ease of modeling:* The rigidity provided by the relational databases offers reliable data models. However, this rigidity can provide difficulties when it comes to scalability. In this case, having a flexible data model brings more advantages.

4) *Data management:* In the management of the data, we aim to utilize a storage solution that provides adequate tools for managing a database, such as data recovery and data security.

5) *Pipeline integration:* The integration of a database can be crucial when it comes to a data pipeline. The database should have support for drivers in different programming languages.

Table 2. Fields of the data after the transformation stage.

Field Name	Description	Data Type
ID	A unique identifier string for each startup object	String, Null
Company name	The name of the startup company in string format	String, Null
Logo	The URL that contains the logo of the startup	String, Null
Location country	The country where the startup is located (e.g. Türkiye)	String, Null
Location city	The city where the startup is located. (e.g. Istanbul)	String, Null
Field of operation	The field that the startup operates (e.g. Technology)	String, Null
Website	The website URL of the startup company	String, Null
LinkedIn URL	The LinkedIn profile URL of the company	String, Null
Employee count	Employee count of the startup	String, Null
Establishment date	The establishment date of the startup	String, Null
Description	The provided description of the startup	String, Null
Research fields	Research fields of the startup	String, Null

With regards to these criteria, MongoDB was selected as the storage solution as it conforms to all the needs of the system. MongoDB is an open-source, document-based NoSQL database [63]. It provides high performance with high availability and automatic scaling. It provides fast and scalable data storage for semi-structured data. A data record in MongoDB is called a document and is stored in JSON format, and a group of data is called a collection, the equivalent of tables in relational databases. A field in a document can contain other documents, arrays, and arrays of documents. [76] With this feature, it's possible for MongoDB to support different data structures and formats, helping with scalability. While setting up a MongoDB instance, different roles are

created for different users accessing the database to ensure access control in the database. It is possible to query, search, and analyze the data within a collection.

The underlying infrastructure of data storage is another important decision to make when setting up the data storage application. There are different options when it comes to infrastructure, it can be done by hosting the service in a dedicated system (such as a physical server) or using a cloud-based solution. In recent years, cloud infrastructures are getting more popular among data storage systems for their various benefits [83]. Firstly, it allows easy set up for the users as it erases the necessity of employing dedicated hardware to host the service; additionally, the cloud infrastructure can be easily scaled horizontally when it's required, which is in line with the requirements set in the storage strategy section. And finally, any data loss that can be caused by a hardware failure is avoided since the storage is not bound to one single point of failure. [23] However, despite these advantages, cloud infrastructure is generally used with a cloud service provider (such as Amazon Web Services, Google Cloud Services etc.) and therefore, these services require a monthly fee for the users. For these reasons, cloud services can be costly for the users if utilized without caution. Yet, despite the cost disadvantage, for the scope of this study, the advantages have outweighed the disadvantages, leading us to choose a cloud infrastructure to host the MongoDB cluster.

D. Data Preprocessing and Descriptive Analysis

The data stored in the MongoDB collection is analyzed to gain valuable insights. As outlined in the Data Preprocessing and Descriptive Analysis section of chapter II, the necessary data preparation steps are applied before the analysis. A Python script developed within Jupyter Notebook [84], a web-based coding environment, facilitates this process. Using the pymongo driver, the script accesses the MongoDB instance and downloads the startup data. For data analysis, the Python libraries "Pandas" [85] and "Matplotlib" [86] are employed to handle and visualize the data effectively. In the MongoDB collection, there is currently data for 4319 startups.

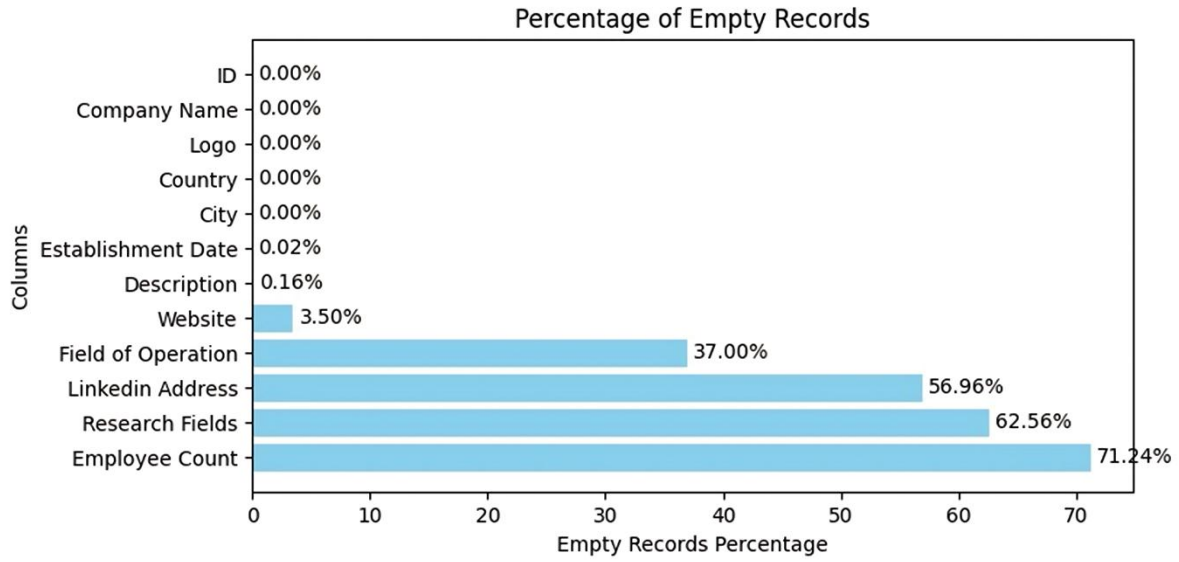


Figure 4. Information on data attributes.

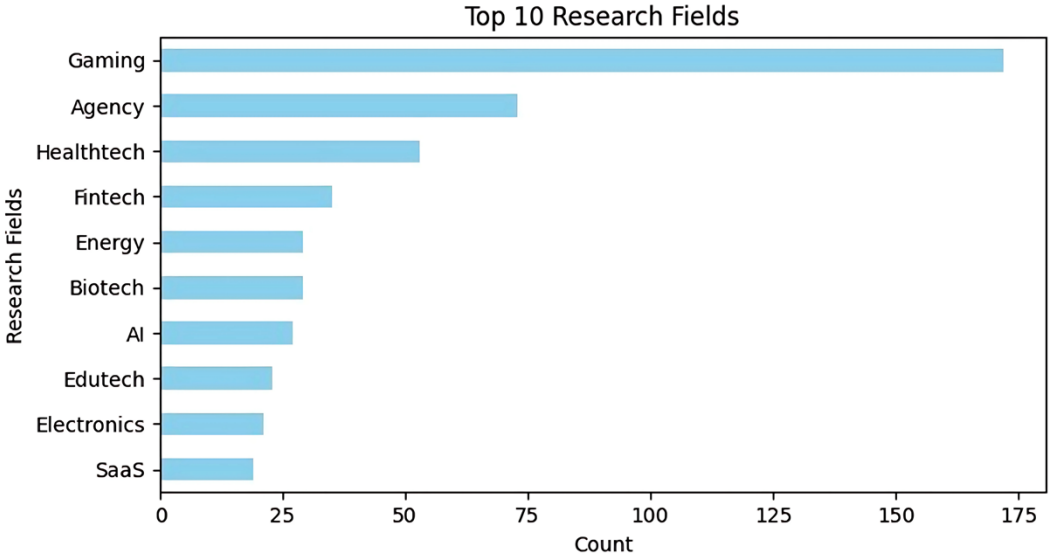


Figure 5. Top 10 research fields

Initially, the dataset was examined for missing or empty values to ensure completeness and integrity of the data before proceeding with further analysis. Some data structure corrections were made during this examination. For example, the data received in Turkish, field of operation and research fields columns, were translated into English for language integrity

In Figure 4 Percentage of Empty Records graph shows significant gaps, particularly in the startups' research fields and LinkedIn addresses. These gaps are essential for the integrity of the analysis. When we draw a bar chart of the data we have in the research fields, we obtain the Top 10 Research Fields graph in Figure 5. When we examine the field of operation information, the Top 10 Research Fields graph of Figure 6 shows that the most studied fields are "SaaS/PaaS and Marketplace". In the Top 10 City in Türkiye graph of Figure 7, we can see the information in which cities the startups are most established in Türkiye.

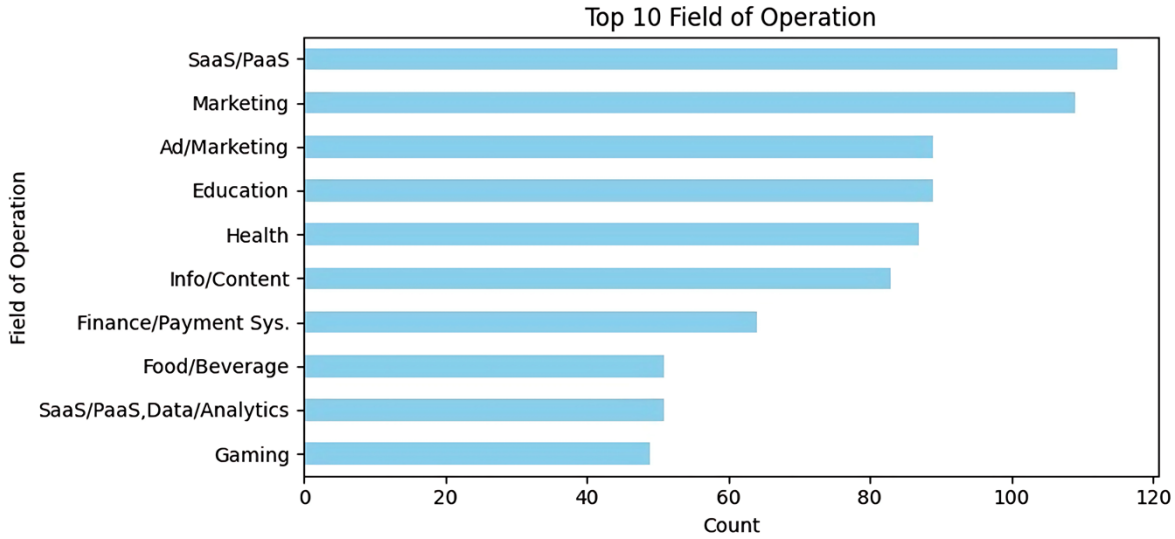
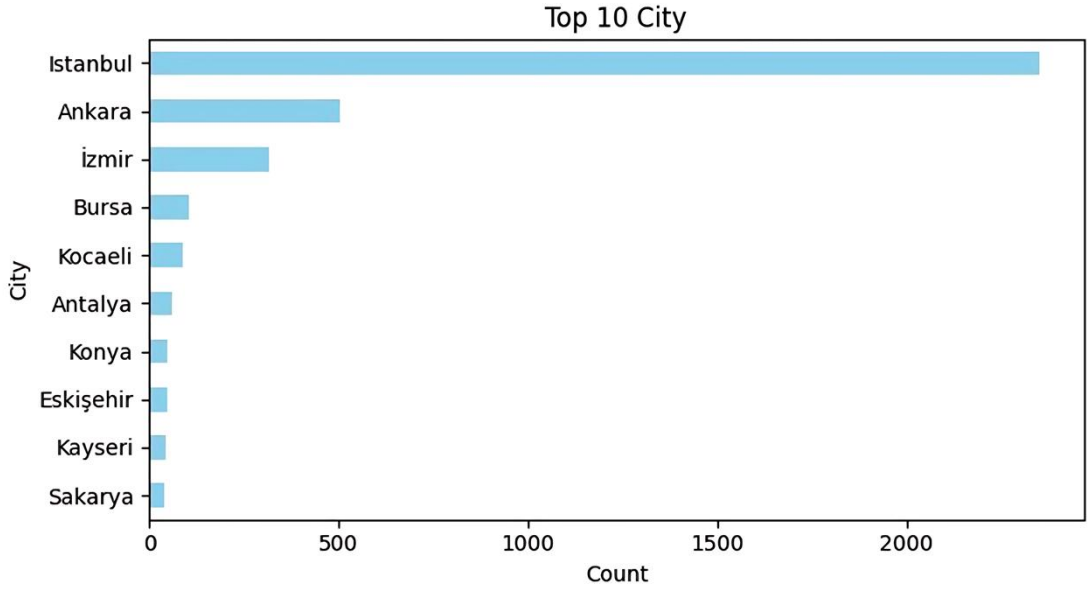


Figure 6. Top 10 field of operation

Figure 7. Top 10 city in Türkiye

Table 3 shows the descriptive statistics results of the first 10 most common fields of operation in table three. Accordingly, we observe that the median and mode values of SaaS & PaaS are the same, and the mean value is very close to them. In this case, the number of employees in companies under the SaaS & PaaS field of operation may show a symmetric distribution model. The fact that the standard deviation is still very high from the mean indicates that there may be a large variety between employee counts. In other words, while there are many employees in some companies under the SaaS & PaaS, there may be fewer employees in some companies. The fact that quartile one is 3 and quartile three is 10, but the mean value is 12, gives the assumption that the maximum value here is that there are some companies with a very high employee count. When we look, we see that the median and mode values for Information & Content, Advertising & Marketing, and Social Enterprise and Entertainment are the same, but their minimum values differ greatly, and when we observe that quartile 1 and quartile 3 are 3 and 10, the fact that the mean value is very far from the median and mode suggests that they may be tailed distributions. When we consider the number of employees in companies, we see that standard sales are much larger than both the mean, median, and mode. So, there is great diversity in terms of the number of company employees. In other words, it seems that there are no similar numbers of employees in companies in the same sector. When we look at the Data & Analytics and Education fields here, we can say that they are also close to SaaS & PaaS. On the other hand, we can think that the Marketplace is generally different from all of them. Because the mode and median values are not the same, the mean value is not similar either. The standard deviation is again very high; in order to make a more detailed comment, we can examine the box plots of those number vectors for



the field of operation from which we extracted this descriptive analysis. According to Figure 4, the number of employees of companies operating in these sectors do not seem to be symmetrical or normal distribution.

Table 3. Statistics of top 10 field of operations with highest employee count.

Field of operation	Mean	Median	Mode	STD	Q1	Q3	IQR
SaaS & PaaS	12.1295	10	10	15.6028	3	10	7
Marketplace	9.2697	6.5	3	12.3081	3	10	7
Data & Analytics	12.2437	10	3	16.6329	3	10	7
Education	11.4795	10	3	14.8599	3	10	7
Information & Content	7.6574	3	3	10.6916	3	10	7
Advertising & Marketing	8.3147	3	3	10.2853	3	10	7
Social Enterprise	10.1226	3	3	16.2678	3	10	7
Entertainment	8	3	3	10.5606	3	10	7
Finance & Payment Systems	9.5151	10	3	11.5762	3	10	7
Hardware	11.3913	10	3	19.6418	3	10	7

There are some anomalous companies for all of them with very high employee numbers (100). The Hardware sector is interesting; it has a relatively more balanced distribution than the anomalous with 100 employees. The number of employees in the Information & Content, Advertising & Marketing, and Entertainment sectors varies by company but is generally low. However, there are some anomalous companies with high employee numbers. In other sectors, the number of employees is relatively higher.

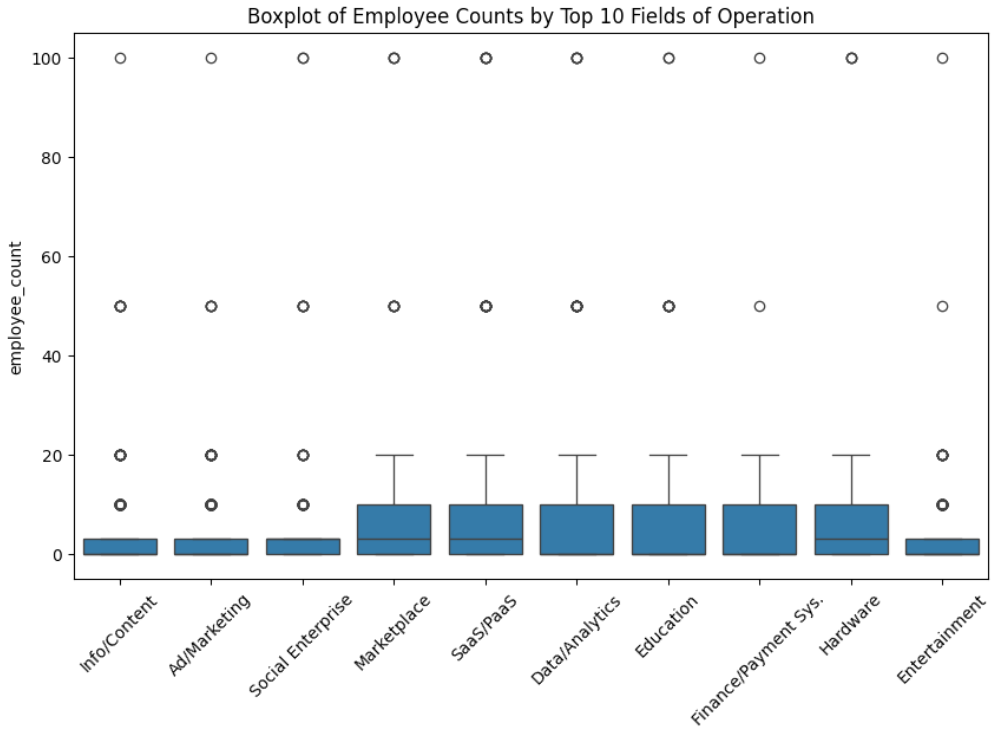


Figure 8. Boxplot of Employee Counts by Top 10 Fields of Operation

For finance and payment systems, while the median and mean are close, the mode value is very far, so we can say that there is not much balance in the number of employees. In order to say in which direction, the imbalance is, we need to look at the distribution or box plots. According to Figure 8, the distribution is in the following direction.

V. CONCLUSION

In this study, a robust and analytically serviceable integrated database for the Turkish startup ecosystem has been successfully created by utilizing advanced web scraping techniques, ETL processes powered and transformed by modern data engineering solutions, and document-based NoSQL database infrastructure. For the processing and storage of comprehensive, up-to-date, and dynamic data belonging to startups, MongoDB has been identified as the most suitable for the data format at hand and stands out with its flexibility and scalability.

By applying exploratory data analysis methodologies on the data collected on a central database at the end of a detailed data pre-processing and integration process, critical insights such as geographical distribution, fields of activity and workforce metrics of the Turkish startup ecosystem were also revealed. In addition, the findings obtained as a result of the experiments conducted within the scope of the study also emphasize the importance of a meticulously designed data pipeline in order to generate actionable insights. For the integrated database to serve with complete and accurate information in all areas and to further enrich the analytical capabilities of the database, it would be good to add additional dimensions such as financing and growth data and sectoral impact to the collected startup data. Since MongoDB has a flexible data schema structure, such dimension additions to the collections on the integrated database can be easily handled. The methodology in this study can be integrated into any field. It is believed that these processes, which are considered as a continuation of the study, will open the door to different research areas.

A robust and reliable data storage for up-to-date startup data has the potential to provide valuable insights to many shareholders of the startup ecosystem. The framework we have provided has the potential to not only collect this data but also ensure data quality with high availability. In conclusion, this study provides a strong foundation for a data-driven investigation of the startup environment in Türkiye and offers valuable methods that can be adapted to other fields.

REFERENCES

[1] Startups Watch. (2024). Startups Watch. İstanbul, https://startups.watch/, (24.10.2024).

[2] Startup Market. (2024). Startup Market. İstanbul, https://startupmarket.co/, (31.10.2024).

[3] Buyukbalci, P., Sanguineti, F., & Sacco, F. (2024). Rejuvenating business models via startup collaborations: Evidence from the Turkish context. *Journal of Business Research*, *174*(8), 114521.

[4] Sakarya, Ş., & İlkdoğan, S. (2023). Türkiye'de Startup Yatırımları ve Finansmanı. *Bucak İşletme Fakültesi Dergisi*, *6*(2), 146-171.

[5] Eroglu, Y., & Rashid, L. (2022). The impact of perceived support and barriers on the sustainable orientation of Turkish startups. *Sustainability*, *14*(8), 4666.

[6] Birden, M., & Bastug, M. (2020). The Impact of Incubators on Entrepreneurial Process in Turkey: A guide for Startups. *Journal of Business Economics and Finance*, *9*(2), 132-142.

[7] Knight, A., Greer, L. L., & De Jong, B. (2020). Start-up teams: a multidimensional conceptualization, integrative review of past research, and future research agenda. *Acad. Manag. Ann.*, *14*(1), 231–266.

[8] Startup Genome LLC (2022). The global Startup ecosystem report GSER 2024. San Fransisco, https://startupgenome.com/article/global-startup-ecosystem-ranking-2024-top-40, (18.11.2024).

[9] Mandel, M. (2017). How the Startup Economy is Spreading Across the Country. And How It Can Be Accelerated. Washington, Progressive Policy Institute, https://www.progressivepolicy.org/how-the-startup-economy-is-spreading-across-the-country/, (18.11.2024)

[10] Basole, R. C., Russell, M. G., Huhtamäki, J., Rubens, N., Still, K., & Park, H. (2015). Understanding business ecosystem dynamics: A data-driven approach. *ACM Transactions on Management Information Systems (TMIS), 6*(2), 1-32.

[11] Ziakis, C., Vlachopoulou, M., & Petridis, K. (2022). *Start-up ecosystem (StUpEco):* A conceptual framework and empirical research. *Journal of Open Innovation: Technology, Market, and Complexity*, *8*(1), 35.

[12] Jáki, E., Molnár, E. M., & Kádár, B. (2019). Characteristics and challenges of the Hungarian startup ecosystem. *Vezetéstudomány-Budapest Management Review, 50*(5), 2-12.

[13] Türkiye Technohub Platformu. (2024). Türkiye Technohub Platformu. Ankara, https://turkiyetechnohub.org, (05.09.2024)

[14] Massimino, B. (2016). Accessing online data: Web-crawling and information-scraping techniques to automate the assembly of research data. *Journal of Business Logistics*, *37*(1), 34-42.

[15] Mancosu, M., & Vegetti, F. (2020). What you can scrape and what is right to scrape: A proposal for a tool to collect public Facebook data. *Social Media+ Society*, *6*(3), 2056305120940703.

[16] Krotov, V., Johnson, L., & Silva, L. (2020). Tutorial: Legality and Ethics of Web Scraping. *Communications of the Association for Information Systems*, *47*, 539-563.

[17] Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, *13*(3), 145-168.

[18] A. W. Sudrajat, Ermatita & Samsuryadi (2023), Extending The Data Integration Model As The Foundation Of Business Intelligence: A Systematic Literature Review. *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI),* 20-21 September 2023*,* 175-182.

[19] Agrawal, P. (2023). Web Scraping and its Applications. *International Journal of Scientific Research in Engineering And Management*, *7*(10), 1-11.

[20] Luscombe, A., Dick, K., & Walby, K. (2021). Algorithmic thinking in the public interest: navigating technical, legal, and ethical hurdles to web scraping in the social sciences. *Quality & Quantity, 56*, 1023-1044.

[21] Goulas, S., & Karamitros, G. (2024). How to harness the power of web scraping for medical and surgical research: An application in estimating international collaboration. *World journal of surgery, 48*(6)*,*1297-1300.

[22] Rodrigues, L. A., & Polepally, S. K. (2021). *Creating Financial Database for Education and Research: Using WEB SCRAPING Technique*. Master thesis, Dalarna University, School of Technology and Business Studies, Dalarna.

[23] Styawati, A. Nurkholis, F. A. Ans, S. Alim, L. Andraini & R. A. Prasetyo. (2023) Web Scraping for Summarization of Freelance Job Website Using Vector Space Model. *2023 IEEE 9th Information Technology International Seminar (ITIS)* 18-20 October 2023, 1-5.

[24] Barba, G., Lazoi, M., & Lezzi, M. (2024). Bibliometric Insights into Web Scraping and Advanced AI-Based Models for Valuable Business Data. *ICEIS*, 1, 321-328.

[25] Dong, H., Zhang, C., Li, G., & Zhang, H. (2024). Cloud-native databases: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *36*(12), 7772-7791.

[26] Zhou, D., Yan, Z., Fu, Y., & Yao, Z. (2018). A survey on network data collection. *Journal of Network and Computer Applications*, *116*, 9-23.

[27] Spaniol, M., Denev, D., Mazeika, A., Weikum, G., & Senellart, P. (2009). Data quality in web archiving. *In Proceedings of the 3rd Workshop on Information Credibility on the Web* 20 April 2009, 19-26.

[28] Vording, R. M. (2021). *Harvesting unstructured data in heterogenous business environments; exploring modern web scraping technologies.* Bachelor's thesis, University of Twente, Enschede.

[29] Gandhi, R., Khurana, S. & Manchanda, H. (2023). ETL Data Pipeline to Analyze Scraped Data. *Decision Intelligence, Proceedings of the International Conference on Information Technology, InCITe 2023, Volume 1,* 379-388.

[30] Simitsis, A., Skiadopoulos, S., & Vassiliadis, P. (2023, March). The History, Present, and Future of ETL Technology. *Proceedings of the 25th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP) co-located with the 26th International Conference on Extending Database Technology and the 26th International Conference on Database Theory (EDBT/ICDT 2023)* 28 March 2023, 3-12.

[31] Singhal, B., & Aggarwal, A. (2022, December). ETL, ELT and reverse ETL: a business case study. *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)* 16-17 December 2022, 1-4.

[32] Raj, A., Bosch, J., Olsson, H. H., & Wang, T. J. (2020, August). Modelling data pipelines. *2020 46th Euromicro conference on software engineering and advanced applications (SEAA)* 26-28 August 2020, 13-20.

[33] Walha, A., Ghozzi, F., & Gargouri, F. (2024). Data integration from traditional to big data: main features and comparisons of ETL approaches. *The Journal of Supercomputing*, *80*(19), 26687-26725.

[34] Nwokeji, J. C., & Matovu, R. (2021). A systematic literature review on big data extraction, transformation and loading (etl). *Intelligent Computing: Proceedings of the 2021 Computing Conference* 15-16 July 2021, 308-324.

[35] Bhatlawande, S., Rajandekar, R., & Shilaskar, S. (2024). Implementing Middleware Architecture for Automated Data Pipeline over Cloud Technologies. *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)* 06-07 April 2024, 506-513.

[36] Hafyani, H., Abboud, M., & Taher, Y. (2021). A Microservices Based Architecture for Implementing and Automating ETL Data Pipelines for Mobile Crowdsensing Applications. *2021 IEEE International Conference on Big Data (Big Data)* 15-18 December 2021, Orlando, FL, USA, 5909-5911.

[37] Singu, S. K. (2021). Designing scalable data engineering pipelines using Azure and Databricks. *ESP Journal of Engineering & Technology Advancements*, *1*(2), 176-187.

[38] Diouf, P. S., Boly, A., & Ndiaye, S. (2018). Variety of data in the ETL processes in the cloud: State of the art. *2018 IEEE International Conference on Innovative Research and Development (ICIRD)* 11-12 May 2018, Bangkok, Thailand, 1-5.

[39] Loshin, D. (2010). The practitioner's guide to data quality improvement. *Elsevier*.

[40] Lamer, A., Saint-Dizier, C., Paris, N., & Chazard, E. (2024). Data Lake, Data Warehouse, Datamart, and Feature Store: Their Contributions to the Complete Data Reuse Pipeline. *JMIR medical informatics*, *12*, e54590.

[41] Makris, A., Tserpes, K., Spiliopoulos, G., Zissis, D., & Anagnostopoulos, D. (2021). MongoDB Vs PostgreSQL: A comparative study on performance aspects. *GeoInformatica*, *25*, 243-268.

[42] Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A. M., & Luo, B. (2023). SQL and NoSQL database software architecture performance analysis and assessments—a systematic literature review. *Big Data and Cognitive Computing, 7*(2), 97.

[43] Ali, A., Naeem, S., Anam, S., & Ahmed, M. M. (2023). A state of art survey for big data processing and nosql database architecture. *International Journal of Computing and Digital Systems*, *14*(1), 1-1.

[44] Ambre, A., Gaikwad, P., Pawar, K., & Patil, V. (2019). Web and android application for comparison of e-commerce products. *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, *5*(4), 266-268.

[45] Rathore, M., & Bagui, S. S. (2024). MongoDB: Meeting the Dynamic Needs of Modern Applications. *Encyclopedia*, *4*(4), 1433-1453.

[46] Ereth, J. (2018). DataOps-Towards a Definition. *Lernen. Wissen. Daten. Analysen. (LWDA 2018)* August 22–24 2018, Mannheim, Germany, 104-112.

[47] Bergh, C., Benghiat, G., & Strod, E. (2019). *The DataOps cookbook (Second Version)*. DataKitchen Hqrs. https://www.devopsschool.com/blog/wp-content/uploads/2021/07/DK_dataops_book_2nd_edition.pdf

[48] Boegershausen, J., Datta, H., Borah, A., & Stephen, A. T. (2022). Fields of gold: Scraping web data for marketing insights. *Journal of Marketing*, *86*(5), 1-20.

[49] Vanden Broucke, Seppe, & Bart Baesens (2018). Practical Web scraping for data science. New York, NY: Apress.

[50] Henrys, K. (2021). Importance of web scraping in e-commerce and e-marketing. *SSRN Electron. Journal*.

[51] Barbera, G., Araújo, L.F., & Fernandes, S.C. (2023). The Value of Web Data Scraping: An Application to TripAdvisor. *Big Data Cogn. Comput*., *7*, 121.

[52] Ticu, C. C. (2021). *The Austrian start-up incubator ecosystem: A web scraping, AWS ML & text analytics competitor analysis on digital content*. Master's thesis, Central European University, Department of Economics and Business, Vienna.

[53] Vassiliadis, P., Simitsis, A., & Baikousi, E. (2009). A taxonomy of ETL activities. *DOLAP '09: Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP* 2009, 25-32.

[54] Richardson, L. (2024). Beautiful Soup Documentation. Cambridge, https://www.crummy.com/software/BeautifulSoup/bs4/doc/#, (05.09.2024).

[55] Google Developers. (2024). Puppeteer Documentation. Mountain View, https://pptr.dev/guides/what-is-puppeteer, (03.09.2024).

[56] Ali, S., & Wrembel, R. (2017). From conceptual design to performance optimization of ETL workflows: current state of research and open problems. *The VLDB Journal*, *26*, 777 - 801.

[57] Rajić, M.N., Milosavljević, P., & Kostić, Z. (2023). Knowledge and Data Management: The Cornerstone of Effective Organizational Strategy. *2023 International Conference on Big Data, Knowledge and Control Systems Engineering (BdKCSE)* 2-3 November 2023, Sofia, Bulgaria, 1-7.

[58] Shrestha, L., & Sheikh, N. (2022). Multiperspective Assessment of Enterprise Data Storage Systems: Literature Review. *2022 Portland International Conference on Management of Engineering and Technology (PICMET)* 07-11 August 2022, Portland, OR, USA, 1-8.

[59] Nambiar, A., & Mundra, D. (2022). An overview of data warehouse and data lake in modern enterprise data management. *Big data and cognitive computing*, *6*(4), 132.

[60] Khine, P. P., & Wang, Z. S. (2018). Data lake: a new ideology in big data era. *ITM Web of Conferences* May 4-6 2018, Girne, Turkey, 03025.

[61] Nurhadi, Kadir, R. B. A., & Surin, E. S. B. M. (2021). Evaluation of NoSQL Databases Features and Capabilities for Smart City Data Lake Management. *In Information Science and Applications: Proceedings of ICISA 2020*, Singapore 16-18 December, Bali, Indonesia, 383-392.

[62] Koutroumanis, N., Kousathanas, N., Doulkeridis, C., & Vlachou, A. (2021). Declarative Querying of Heterogeneous NoSQL Stores. *SEA-Data@ VLDB* 20 August 2021, Copenhagen, Denmark, 42-43.

[63] Niu, J., Xu, J., & Xie, L. (2018). Hybrid Storage Systems: A Survey of Architectures and Algorithms. *IEEE Access*, *6*, 13385-13406.

[64] Davoudian, A., Chen, L., & Liu, M. (2018). A Survey on NoSQL Stores. *ACM Computing Surveys (CSUR), 51*, 1-43.

[65] Membrey P, Plugge E, Hawkins T & Hawkins D. (2010). The definitive guide to mongoDB: the noSQL database for cloud and desktop computing. *Springer, Berlin.*

[66] Rathore, M., & Bagui, S. S. (2024). MongoDB: Meeting the Dynamic Needs of Modern Applications. *Encyclopedia*, *4*(4), 1433-1453.

[67] Alghamdi, T. A., & Javaid, N. (2022). A survey of preprocessing methods used for analysis of big data originated from smart grids. *IEEE Access*, *10*, 29149-29171.

[68] Famili, A., Shen, W. M., Weber, R., & Simoudis, E. (1997). Data preprocessing and intelligent data analysis. *Intelligent data analysis*, *1*(1), 3-23.

[69] Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91-99.

[70] Zhu, X., Wu, X., & Chen, Q. (2003). Eliminating class noise in large datasets. *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* August 21-24 2003, Washington, DC, USA, 920-927.

[71] Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *IEEE Access*, *8*, 54776-54788.

[72] Katya, E (2023). Exploring Feature Engineering Strategies for Improving Predictive Models in Data Science. *Research Journal of Computer Systems and Engineering*, *4*(2), 201–215.

[73] Abt, K. (1987). Descriptive data analysis: a concept between confirmatory and exploratory data analysis. *Methods of information in medicine*, *26*(02), 77-88.

[74] Ali, Z., Bhaskar, S. B., & Sudheesh, K. (2019). Descriptive statistics: Measures of central tendency, dispersion, correlation and regression. *Airway*, *2*(3), 120-125.

[75] Blanca, M. J., Arnau, J., López-Montiel, D., Bono, R., & Bendayan, R. (2013). Skewness and kurtosis in real data samples. *Methodology*, *9*(2), 78-84.

[76] Kim, H. Y. (2013). Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. *Restorative dentistry & endodontics*, *38*(1), 52-54.

[77] Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, *126*(5), 1763-1768.

[78] Jose, B., & Abraham, S. (2017). Exploring the merits of nosql: A study based on mongodb. 2017 International Conference on Networks & Advances in Computational Technologies (NetACT) 20-22 July 2017, Thiruvananthapuram, India, 266-271.

[79] Türkiye Technohub Platformu. (2024). Türkiye Technohub Platformu. Ankara, https://turkiyetechnohub.org/, (28.08.2024).

[80] Requests: HTTP for Humans. https://requests.readthedocs.io/en/latest/ (05.09.2024)

[81] Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., & Stafford, R. (2003). Data transfer object. Patterns of Enterprise Application Architecture. Addison Wesley, 347-356.

[82] Rathore, M., & Bagui, S. S. (2024). MongoDB: Meeting the Dynamic Needs of Modern Applications. *Encyclopedia*, *4*(4), 1433-1453.

[83] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bernstein, P. A., ... & Suciu, D. (2022). The Seattle report on database research. *Communications of the ACM*, *65*(8), 72-79.

[84] Jupyter Team. Jupyter Notebook Documentation. https://jupyter-notebook.readthedocs.io/en/latest/ (08.09.2024)

[85] Pandas Development Team. Pandas Documentation. https://pandas.pydata.org/docs/ (16.09.2024)

[86] Matplotlib Development Team. Matplotlib Documentation. https://matplotlib.org/ (16.09.2024)