School of Information Technology and Engineering at the ADA University

School of Engineering and Applied Science at the George Washington University

GRAPH-BASED VISUALIZATION & ANALYSIS OF AZERBAIJANI WEB

A Thesis
Presented to the Graduate Program of Computer Science and Data Analytics.
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
Anar Shikhaliyev

April, 2024

THESIS ACCEPTANCE

This Thesis by: Anar Shikhaliyev

Entitled: Big Graph: Visualization & Analysis of Azerbaijani Web

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

(Adviser)	(Date)
(Program Director)	(Date)
(Dean)	(Date)

ABSTRACT

An in-depth comprehension of the local Azerbaijani internet landscape is essential for examining the patterns of information dissemination and the impact on the local network, as well as assessing Azerbaijan's reliance on foreign sources during cyber-attacks or national crises. In order to enhance this understanding and optimize the acquisition of local data, it is essential to use a web crawler followed by graphical analysis. The objective of this project is to construct a comprehensive network diagram of the Azerbaijani web, and thereafter examine its interconnections and identify the most significant nodes. The objective of this project is to establish a comprehensive list of local websites, design a web crawler to go through each webpage and its external linkages, build a visualization using a graph-based approach to display important information, and use a ranking algorithm to quantify the influence scores. A Python crawler program is created to systematically search and get data from the database of local websites provided by the Ministry of Communication & Information Technologies. The extracted data is saved in both a Postgres database and a Neo4j graph database. The graph is composed of nodes that represent web sites, and edges that reflect relationships between them. A method for page ranking is used to quantify the significance of nodes. The findings mostly focus on visualisation, while some analytical tasks need separate implementation.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
API	Application Programming Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
JSON	JavaScript Object Notation
SQL	Structured Query Language
URL	Uniform Resource Locator
AJAX	Asynchronous JavaScript and XML
GUI	Graphical User Interface
DDoS	Distributed Denial of Service
ToS	Terms of Service

INTRODUCTION

From the beginning, internet has grown quickly and now it is a big store of knowledge about many subjects, languages and cultures. There are billions of webpages saved on World Wide Web which keep important data varying from text material to multimedia resources. The required efficient methods for accessing and analyzing this huge information cannot be stressed enough. The minimum size of indexed World Wide Web is expected to be more than 4.72 billion pages [1]. The main factors pushing up the amount of web content include more online platforms and services, a growing number of users and advances in technology. Worldwide, there are 5.35 billion people using internet and it is predicted that by 2029 this figure will rise to 7.9 billion individuals who use the internet. Typical uses for internet are work-related activities as well as streaming, gaming or simply scrolling through websites. 5.35 billion people, which is more than two-thirds of the world's population (around 8 billion), can use internet now. If internet characteristics remain up-to-date and flexible with the changing world, utilization will increase in large numbers and rapidly. [2]

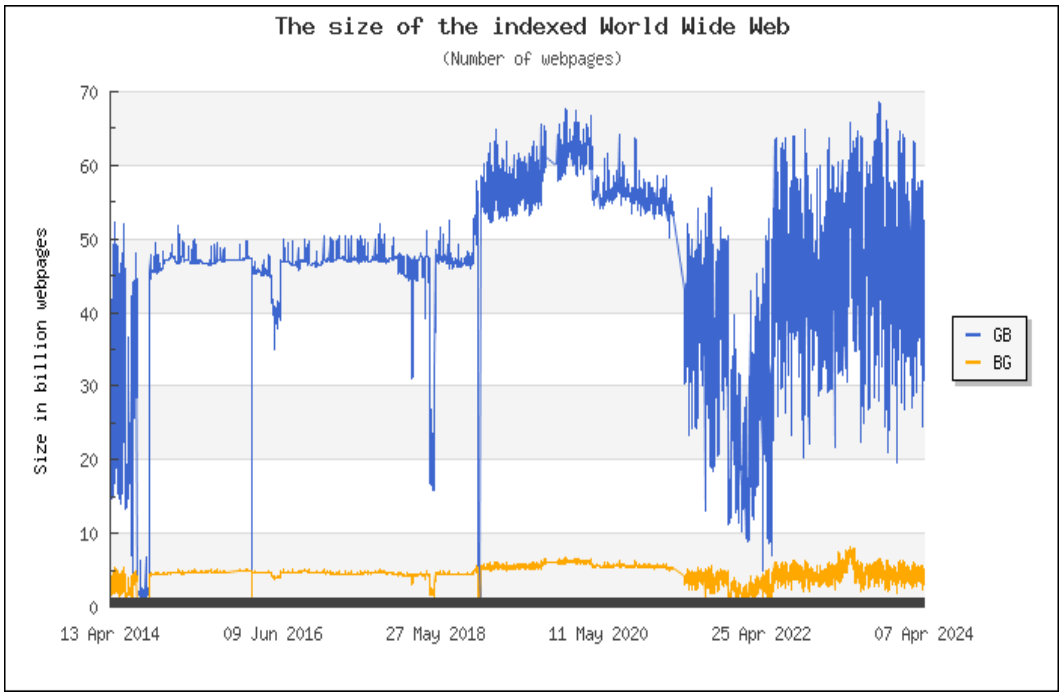

Figure 1: Size of World Wide Web [1]

In the year 2022, there were 8.32 million internet users which is about 81.1 percent of people in Azerbaijan who had access to the internet [3]. The study of the structure and dynamics of local websites becomes very important in our digital world because online information in Azerbaijani language has special importance for us as well.

Information on the web is diverse, comprising of text, photographs, videos and metadata. All these types of web data give important indications about various aspects in society like economy or culture to name just two examples. Analyzing this information can bring out trends, patterns and connections that help with making decisions as well as promoting new ideas while spreading knowledge around. The big amount and different kinds of internet information make it challenging for regular methods to gather data or study them which require specific tools and ways of doing things.

In the time of digital technology, the internet is like a big, messy and always changing group of information. It makes possible sharing knowledge all over the world with its ability to let people communicate and do business globally through webpages that store data on servers for access via internet links. The appearance and simplicity of using websites have changed how people acquire information, participate in commercial activities and communicate on the internet.

A variety of fields can benefit from web data. In marketing, corporations use online data to analyze customer behavior, find target markets and improve their advertising techniques. Web data in finance assists with market analysis, risk assessment, and finding fraud activities. Likewise, web data is beneficial for the same reasons within healthcare: research needs, caring for patients and supporting public health.

The internet and web-based technologies have brought a big revolution in society, changing how people work, interact with each other, and get services. With the rise of platforms for e-commerce, online banking services as well as social media networks; individuals now can access products/services/information from their homes without any difficulty. The process to democratize access has leveled the playing field by empowering people, corporations and governments in similar ways.

Azerbaijan, similar to other nations, is quickly embracing digitalization as a tactic for boosting economic development, stimulating innovation and enhancing public services. With the increasing rates of internet availability and smartphone usage, people in Azerbaijan are becoming more connected and involved in the digital world. The country's strong commitment towards utilizing information technology for its national advancement can be witnessed through governmental actions such as improving digital skills among citizens; expanding broadband networks across all regions; modernizing public administration methods with innovative approaches like e-government initiatives etc.

As Azerbaijan becomes more linked with the global digital economy, it is very important to understand how current online traffic works and what limits it has. For

governments, corporations, and academics alike - analyzing data online helps them in identifying patterns, assessing potential hazards as well as making good choices. By understanding local internet use trends, reliance on external resources and possible vulnerabilities of Azerbaijan's internet activities; we could enhance digital infrastructure, increase cybersecurity measures plus decrease dangers linked to new threats.

1.1 Graph analytics

Graph analytics is a strong method for understanding the complex patterns and connections in web. It is an evolving approach to data analysis, helping firms comprehend complicated relationships of interconnected entity data in a network or graph. The focus mainly lies on the links between two particular items along with overall structural characteristics of the graph. It is useful for many things like looking at social network connections, finding cyber threats and identifying possible customers by shared tastes. Data on social networks, online pages and linkages, road networks, communications networks as well as financial transaction data are a few examples of data that work well on graphs. [4]

1.2 Scraping

Web scraping is a basic method that helps in gathering useful information from the huge amount of non-arranged material available on internet. Online scraping makes it possible to collect, study and understand information on a big scale by automatically collecting data from online sites. Web scraping helps in getting back all types of data like news stories, product lists, social media posts and forum talks for analysis and display purposes.

Before the invention of APIs, web scraping was the only way for a computer to collect details from internet sources. APIs are special interfaces which make it easier for applications and servers to communicate with each other. They are very quick and efficient technologies that give data in an organized format. You can use the API to get different types of information, like articles from Wikipedia or tweets on Twitter. All websites have APIs but not all of them offer free access to their APIs. Moreover, sometimes these APIs might have restrictions in terms of how much data they provide and what type it is. Additionally, a website developer is more concerned with focusing and managing the frontend interface rather than the backend API. In conclusion, trusting only on APIs for internet data is not dependable. It becomes essential to use online scraping methods to ensure that the obtained data matches with people's specific needs.

1.2.1 Web Scraping concepts and techniques

Data is taken from websites using the HTTP protocol, the same way that web browsers use. This can be done by either human browsing or automated methods such as web crawlers. Scraped datasets need good preparation and cleaning up for best use of them.

Methods for web scraping allow users to gather data from multiple websites and bring it together in one database or spreadsheet. This makes the data visible and ready for future use. Web scraping includes making and running two software programs: a crawler and a scraper. The crawler is taking data, and the scraper is pulling from downloaded data. The scraper brings in a particular sort of information (which may be seen as an API), pulls out these bits and saves them into database or file after encoding and structuring following user's instructions on how to do so. This new file can be viewed or used in ways that were not possible with the original way of displaying and accessing internet data. Before this method was developed there existed only one means by which you could see web content-through your browser and even then, it was uncopiable. Web scraping makes it possible to do this task, and you can accomplish it in a short time with an easy script. [5] Specifically, a web scraping application starts by creating an HTTP request for gathering resources from a specific domain. The user has the option of constructing this request either as an URL with GET query or as piece of HTTP message having POST query. Once the website that is being targeted receives and deals with the request properly, it will give back to us the resource we aimed for from its side. The resource might exist in different forms like web pages made using HTML, data feeds in XML or JSON format, and multimedia data such as pictures (photos), sound files (audio) or movies videos. After obtaining the web data, extraction process goes on to parse it into a structure that can be understood and then reformat systematically. In a web scraping application, there are typically two important pieces: one for creating an HTTP request - this could be Urllib2 or Selenium; and another part that can parse and pull-out data from raw HTML code - Beautiful Soup or Pyquery. The module Urllib2 has many functions to manage different parts of an HTTP request, like dealing with authentication, redirections and cookies among others. It is about making requests to web pages and handling responses. Selenium is a tool which wraps around browsers such as Google Chrome or Internet Explorer. This allows people to interact with websites by automating their browsing using programming codes instead of manual input on a keyboard or mouse click on the browser window. Beautiful Soup is designed especially to extract data from HTML and other XML sources. This library provides Python methods that are fast for

looking through, locating and modifying a parse tree. It also offers tools to dissect an HTML document as well as extract certain details with the help of lxml or html5lib.

Beautiful Soup can automatically detect the encoding of the parsing being done and transform it into an encoding that can be understood by the client. Pyquery brings a group of Jquery-like tools for parsing xml documents. It is worth noting, Pyquery only uses lxml to efficiently handle XML processing unlike Beautiful Soup.

Web scraping applications can be divided into categories. For instance, some tools such as Nutch or Scrapy are created to automatically discover the structure of a webpage. On the other hand, there are also those like Import.io which offer an internet-based visual interface that eliminates the requirement for writing web scraping code by hand. Nutch, crafted with the Java programming language, is a tough and flexible web crawler. It can be set up in detail, gather data simultaneously from various sources, follow rules of robots.txt and use machine learning. Scrapy is an efficient system made in Python which enables people to extract data from websites repeatedly. It speeds up the development and growth of large crawling projects. The visual interface for web-based crawling imitates actions a human user would take on a website. The crawler with visual web interface is made to make using a tool for scraping the web easier, enabling those who don't program to pull out contents from online sites. Import.io, a well-known web crawler that can extract data from websites without coding needed. People use this tool to find and change unorganized web pages into an arranged way. Import.io's graphical user interface enables users to train and acquire knowledge on data extraction. The data that has been extracted is saved on a dedicated cloud server and can be exported in many forms such as CSV, JSON, or XML. An online crawler equipped with a visual user interface has the capability to collect and display up-to-date data in real-time using SVG or WebGL technology. However, it would not effectively process a large dataset.

Web scraping serves several purposes, such as extracting contact information, doing price comparisons and tracking price fluctuations, aggregating product evaluations from multiple websites, and compiling real estate listings. Additionally, it may be used to monitor meteorological information and identify alterations on websites. Web data may be integrated with other sources of information. As an example, we may repeatedly get the numerical worth of a stock over a period of time on a limited scale in order to graphically demonstrate its price variation. Collectively, we can use web scraping to gather social media feeds and then analyse public attitudes. This may also aid in identifying powerful individuals. On a

large scale, information from almost all websites is consistently taken to construct Internet search engines like Google Search or Bing Search. [6]

1.3 Graph visualization

A goal of information visualisation is to give ways for changing abstract information, like talks or text descriptions, into visual presentations that help in viewing and handling hidden patterns within the base data sets. Methods of graph visualization are frequently employed when there exist inherent connections among comparable data pieces to increase understanding.[17]

In mathematical structure, a graph

$$G = (V, E)$$

is a formal structure consisting of two sets: V signifies the vertices or nodes within the graph and E represents its edges. Each edge has connection to one or two vertices that are called endpoints.[16]

Diagrams, also known as graphs, are used to illustrate preexisting structures in various fields. For instance, social networks can be shown using a graph where each person in a particular group is depicted as a vertex. The several connections between them are symbolized by an edge or set of edges. In biology and chemistry for example molecular and genetic maps, protein creation pathways - they use graphs frequently too. Graphs are frequently employed in software engineering to visually represent the complex organization of software systems or to demonstrate the inner actions and conditions of compilers. In object-oriented field, we use graphs for showing connections between different classes like UML diagrams. Each hierarchical arrangement can usually be shown as a tree, which is a special type of graph. An instance that could be viewed as a hierarchical tree is the file arrangement in an operating system. The structure of an institute's organization might also be depicted as a hierarchical tree. [17]

Even though graph visualisation methods are extensively used in various application areas, they do have limitations that need to be acknowledged. One problem could be the size of the graph being represented. It is possible to make layouts for very big graphs but usually this causes a decrease in readability, particularly for users who are not well-versed with the topic. This is connected to the limited cognitive capacity of people and the screen space restrictions enforced by visualization technology. Another key worry here is about giving a suitable method that helps user to interact with and navigate within data. The objective of

graph visualisation methods is to improve data comprehension by providing uncomplicated and easily understandable arrangements, along with relevant interaction techniques.[17]

1.4 Search Engine Optimization (SEO):

Search Engine Optimization (SEO) refers to the method of making websites and web pages better for increasing their appearance and position in search engine results pages (SERPs). The understanding of being well-placed in search engines has been grasped by many internet organizations. A report that was published lately exposed majority, with 62% precisely, of those who use search engines will only click on results from first page in SERP (search engine results page). In comparison less than 10% would click anything shown beyond third page. In the present digital world, a big amount of online visits comes from search engines. So it is very important that SEO helps make sure websites can be found easily by people who are looking for them and also seen by those they want to reach [19].

Effective SEO strategies involve various techniques, including on-page optimization, odd-page optimization, keyword analysis, technical optimization and so on.

Researching on keywords finds important words and phrases that users might look for, then use them in a planned way across website content, meta tags, and URLs.

On-page optimization makes every web page better by using the right keywords, making good and interesting content, arranging a structure that is easy for users to understand and follow, as well as applying technical methods for success. [20]

Off-page optimization creates strong backlinks from trustworthy external sites, that might enhance a website's power and trustworthiness for search engines. [20]

Technical optimization makes sure websites are technically good, load quickly, work well on mobile devices and follow search engine rules and top methods. [20]

SEO best practices help improve how frequently search engines display websites in their results. This can increase the number of people visiting these sites and make them more relevant, enhancing online presence and opportunities for success [20].

1.5 Motivation for the study

The internet is growing rapidly, and many types of online information are becoming widely accessible, making it hard to understand the structure and behavior of digital environment in Azerbaijan. As the country's efforts to digitize increase, it becomes even more important to comprehend connections, dependencies as well as weak points within local

online ecosystem. However, current studies on web analysis often focus on global or English-dominant online spaces. This overlooks the intricacies and unique aspects of less-represented languages such as Azerbaijani areas.

This study is inspired by the increasing importance of technology in Azerbaijani culture and the internet's role as a significant platform for sharing information, communication, and conducting business. To make wise decisions, plan strategically and maintain national security it is very important to have an understanding of the domestic online environment in Azerbaijan as this country rapidly embraces digitalization while becoming part of global digital economy.

1.6 Objective for the study

Hence, this project plans to utilize advanced techniques in graph analytics and web scraping for a comprehensive exploration of Azerbaijani digital realm. The aim is to gain important understanding about the composition, movement and robustness of Azerbaijan's online environment by forming a graph depiction of local websites - examining connecting patterns on it as well as evaluating dependence levels within this system.

The main aim is to design a trustworthy method for constructing a graph model that mirrors the Azerbaijani web online. This will involve employing web scraping techniques to collect data from major local websites. By portraying web pages as nodes and hyperlinks as edges, the web graph can offer useful understanding into how connections are formed and arranged within Azerbaijan's online ecosystem.

The study wants to understand the link patterns and movement within the web of Azerbaijan. By creating the web graph, we hope to apply graph analytics tools like centrality measures and community discovery algorithms so as to identify key influencers, theme clusters as well as structural elements in this network.

The study objectives also include evaluating the reliance of the Azerbaijani web on outside resources and modelling situations of severe crises like war or natural catastrophes. Moreover, it aims to assess robustness in online infrastructure from Azerbaijan's viewpoint by looking at how local websites are connected and dependent on foreign domains for providing useful insights for strategic planning.

The project's goal is to use and evaluate page ranking algorithms on the constructed web graph. This effort hopes for a betterment in information finding tools, assisting users with locating appropriate and trustworthy material by measuring the importance and impact of web sites within Azerbaijani online interactions.

The project wants to make our understanding of the Azerbaijani online world better, specifically its effects on social development, digital strength and tactical choices. This will be a helpful addition in web science. The aim is to stimulate teamwork across different fields of study and inspire more exploration in analyzing online information along with its presentation by exchanging research outcomes and thoughts.

1.7 Literature review

Large scale Graph-based visualization has been a topic of much research over the years. Many examples of desktop and web-based visualization applications can be seen, such as Cytoscape, Pajek, Tulip, WiGis etc. [7] analyze thoroughly graph-based visualization techniques and how they are used in various areas like social networks, biological networks and the global internet. The authors study basic principles such as methods for arranging graphs, diagrams showing nodes and links as well as interactive ways of presenting visuals. They highlight their effectiveness at discovering patterns, groups or oddities within complex data sets.

The use of graph-based methods for visualizing online data has allowed researchers to study various aspects like connection architectures, user actions and content changes. A tool called WebGraphViz can help in exploring and examining the link structure of websites using graphs. The application gives a visual way for users to show connections between internet sites, find important central nodes and authoritative sources, as well as study patterns in how information spreads within the web graph.[23]

In another study, [22] explored user navigation patterns on e-commerce platforms by utilizing graph-based visualization. The researchers examined user interactions by presenting them as a directed graph. This gave them the ability to recognize common user paths, along with points where users enter and leave the virtual interface. The outcomes offered significant understanding on patterns of user interaction, which helped direct design decisions for improving their experience.

Research on the study of web data using graphs has looked into different aspects of web architectures, behaviours and changes. PageRank algorithm was created in 1998 and it determines how important online sites are by looking at the structure of web graph as well as number of inbound connections. This method is now a basic part for search engines, having control over where search results rank and impacting how users explore internet pages. [8]

2. Methodology

2.1 Dataset

The first stage of this study was obtaining the dataset, which consists of a compilation of websites given by the Ministry of Communications & Information Technologies of Azerbaijan. This dataset is essential for the later steps of analysing and visualising the online environment in Azerbaijan. When it comes to characteristics of the data, the collection comprises URLs that indicate a range of websites hosted inside the Azerbaijani web domain. The collection includes information about the domain name of the website for each item. In total, the dataset holds 13447 domains.

2.2 Storage

Techniques of web scraping have been used to gather data from the websites mentioned in the dataset. For this research project, two significant databases are included in the storage system: traditional object-relational database PostgreSQL and graph database Neo4j. The current situation of big data requires new analytical skills to make good use of all these statistics available. In past times, when it came to databases, SQL ones such as PostgreSQL were mostly chosen; on other hand newer kinds like graph based one called Neo4j were mainly utilized for examining social network and transportation stats. The comparison of PostgreSQL (using SQL) and Neo4j (using Cypher), in terms of our project, is shown clearly. [9]

The primary reason for selecting PostgreSQL, an open-source relational database management system, is its high level of maturity and stability. It also strictly follows ACID (Atomicity, Consistency, Isolation, Durability) rules.

Data integrity is maintained in PostgreSQL by applying constraints like primary keys, foreign keys and unique constraints. These make sure that stored data stays consistent and dependable. Also, it has strong SQL support which lets users to query and modify structured data effectively. It offers an interface that is easy to use for conducting data analysis and

reporting tasks. PostgreSQL includes scalability functions such as table partitioning and replication, making it possible to handle large datasets and a high volume of transactions efficiently. All these features along with its popularity among active open-source communities guarantee the software's long-term maintenance and stability.

On the other hand, schema rigidity talks about how relational schemas are not flexible and this might make it hard to adjust when data models change or grow. This could require changes in the schema and may limit how adaptable applications can be. Complex queries that involve joining data from multiple sources and performing calculations on large datasets could potentially encounter performance delays. To address this, optimisation techniques like indexing and fine-tuning queries may be necessary.

Significantly, graph data is not nicely supported. Even though PostgreSQL can handle relational data modelling, it lacks embedded features for representing and searching in structures based on graphs. This could limit its usefulness in some types of analytical work. If were to choose to use Postgres for our database, every time we had to build graph in the visualization part all over which increases load on the visualization and causes delay.

In my Postgres database, I am saving the below information and ERD diagram of table can also be viewed in Figure 2:

- id: This gives a special identification to every record in the database. It automatically increases for each new entry and acts as the main key.
- url: The URL of the webpage that was scraped.
- protocol: The protocol used in the URL, such as "http" or "https".
- redirected_url: If the URL was redirected to a different location, this field holds the new URL.
- server: The server software used to host the webpage.
- ip_address: The IP address of the server hosting the webpage.
- domain_name: The domain name of the webpage.
- title: The title of the webpage, typically found within the HTML <title> tag.
- description: The meta description of the webpage, providing a concise summary of its content.
- keywords: Meta keywords associated with the webpage, providing additional context about its content.
- headers: HTTP headers returned by the server when you access this webpage, stored in the form of a JSON object to make it simple for you to fetch and parse them.

- country_iso_code: The ISO country code of the country where the server is located.
- country: The name of the country where the server is located.
- city: The city where the server is located.
- lat: The latitude coordinate of the server's location.
- long: The longitude coordinate of the server's location.
- file_path: The file path where the HTML content of the webpage is saved or stored.
- links: a JSON array that has all the links discovered in the webpage. It helps in analyzing and exploring more about how this website is organized.
- error_type: This field is used to store the type of error that happened during scraping, like timeout or connection error.
- error_message: A descriptive message providing more information about the error encountered during scraping.

Figure 2: ERD diagram of Postgres database

Neo4j is a famous graph database management system, and it was welcomed because of its natural capacity to manage graph data structures, as well as strong querying features. We spoke about how scaling horizontally with relational databases might present difficulties. It could lead to less-than-ideal performance when working big datasets due to limits in handling high-volume read and write requests effectively. The birth of NoSQL databases came from this problem, making them more liked by people. NoSQL databases stick to the CAP (Consistency, Availability, and Partition Tolerance) theorem. They are especially good for network partitioning because of their ability to scale horizontally. Graph databases are useful when you have data that changes a lot and is difficult to fit into a pre-defined structure. AllegroGraph, Datastacks and Neo4j are types of NoSQL database which store data in graph style where nodes represent data while relationships show connections between these nodes. These databases, they stress on the interconnections between nodes. In property graph databases such as Neo4j, every node and relationship can possess attributes.

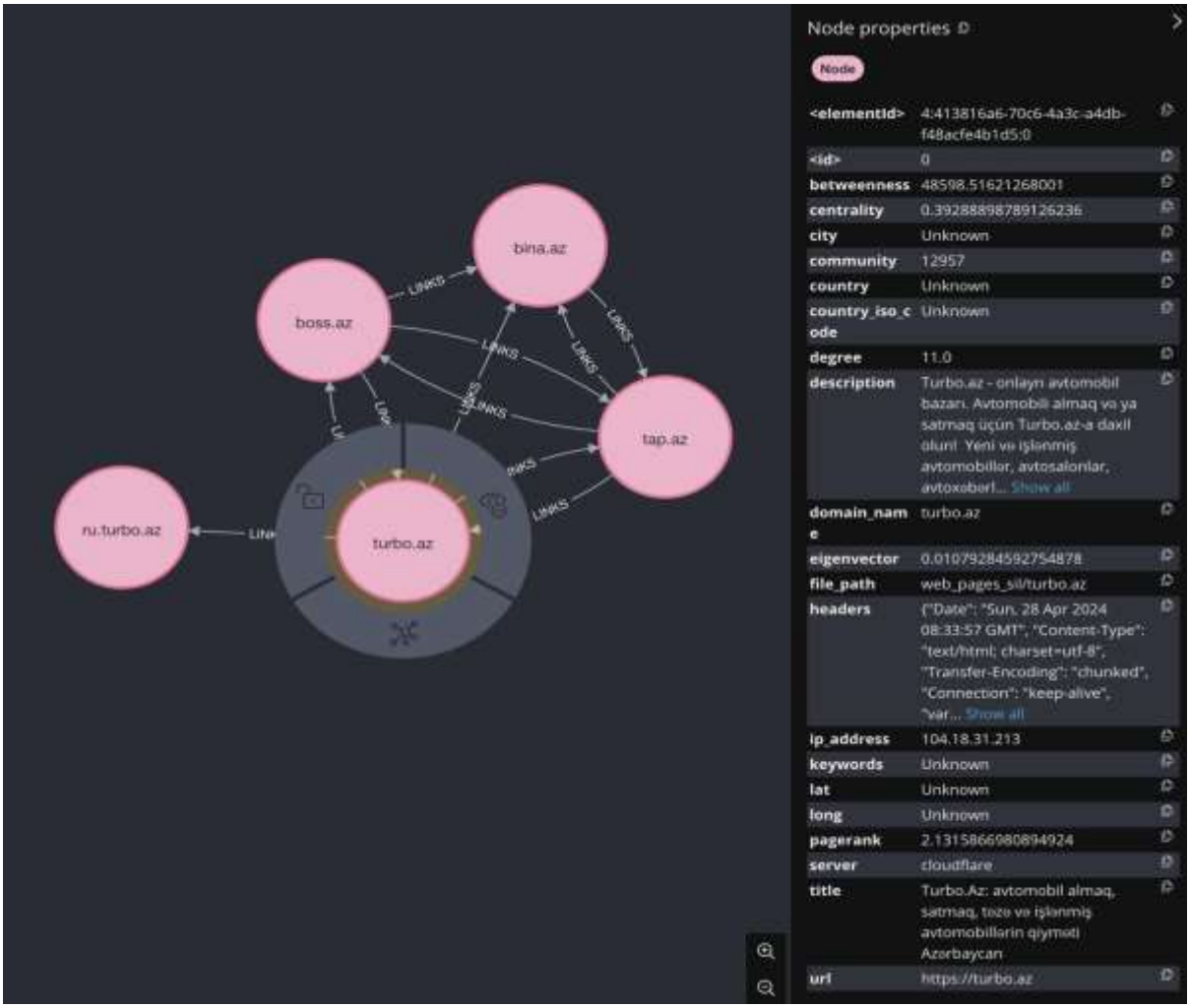

Figure 3: Subgraph to understand Neo4j ways.

2.3 Graph Analytics

2.3.1 Centrality Measures

- Betweenness Centrality

Betweenness centrality is a measure of how much a node controls the flow of information within a network, and it is frequently used to find nodes that act as bridges connecting various parts of that network.

In the algorithm, we calculate the shortest paths connecting all pairs of nodes in a graph. Each node's score is dependent on how many shortest routes pass through it. So, nodes with higher betweenness centrality scores are those that frequently appear on the shortest pathways between other nodes.

Betweenness centrality can be used on graphs that either lack weights or possess only positive weights. The GDS function is built from Brandes' heuristic technique for unweighted graphs. Many concurrent Dijkstra algorithms need to be used in handling weighted graphs, which we applied on our graph. The process requires O(n + m) space and works in O(n * m) time, where n is the number of nodes and m is the number of connections in the graph. [10]

In the graph visualisation that is implemented, betweenness centrality functions as a crucial measure for node colouring. This metric provides useful details about the structural significance of each node within the network (Figure 4).

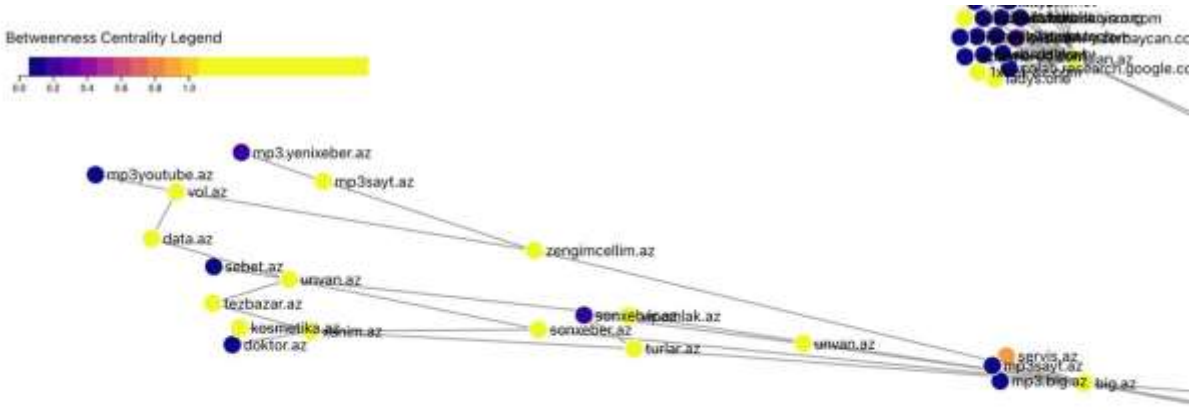

Figure 4: Betweenness centrality represented with color as a differentiating factor in graph

- Eigenvector Centrality

The method of Eigenvector Centrality is a way to compute node influence in a network by considering both direct and indirect impacts. This method assumes that connections from nodes with higher scores will have more impact on the score of a given node, compared to connections from those with lower scores. When the eigenvector score for a specific node is high, it means this particular node has strong relationships to other nodes in the network which also possess high scores.

The process computes the eigenvector of highest absolute value eigenvalue. It uses power iteration procedure to calculate the eigenvalue. In every cycle, it finds centrality score for each node depending on scores from its incoming neighbors. In every power iteration step, the eigenvector gets normalized through L2-norm, which makes normalized outputs the natural result.

The method of PageRank is based on Eigenvector Centrality, with a modification that adds one more opportunity for jumping to another page. [11]

- Degree Centrality

The technique of Degree Centrality is useful for finding nodes in a network that have many connections. Degree centrality gives a number to the amount of links, coming or going, which a node has based on how the relationship projection is directed. For more details about relationship orientations, please see the section discussing syntax for projecting relationships.

This can be utilized for weighted graphics also. In the case of weights, this method sums up all the positive weights of neighboring connections for every node in the network. We ignore any weight that is equal to or less than zero.

The method can be applied to graphs having diverse kinds of relationships, but it will not calculate the degree centrality for every type of link. Rather, the algorithm will treat the network as homogeneous since this is what its characteristics imply. [12]

- ArticleRank Centrality

ArticleRank, which is a variation of the Page Rank algorithm, measures the indirect influence of nodes. The basic principle behind the Page Rank algorithm is that connections from nodes with less connectivity have more importance than those coming from well-connected ones. In this method, Article Rank cuts down on the effect of low-degree nodes by reducing scores sent to their neighbor nodes in every iteration.

The Article Rank of a node v at iteration i is precisely defined as:

$$ArticleRank_i(v) = (1-d) + d \sum_{w \in N_{in}(v)} \frac{ArticleRank_{i-1}(w)}{|N_{out}(w)| + \overline{N_{out}}}$$

Figure 5: Article Rank calculation

- The term "Nin(v)" refers to the incoming neighbours of node v, whereas "Nout(v)" refers to the outbound neighbours of node v.
- The damping factor, denoted as d, is a value that ranges between 0 and 1.
- Nout is the mean number of outgoing edges from a node.[13]

Community detection

We applied community detection methods to identify communities in the Azerbaijani online graph. These are groups of web sites that show strong connections with each other, either because they have similar characteristics or deal with related subjects. The results from community detection provide important understanding about how web graph is fundamentally structured and organized, revealing patterns of cohesion and modularity.

Methods of community identification are employed to evaluate the clustering or partitioning of groups of nodes, along with their propensity to strengthen or disintegrate.

The Louvain method, an algorithm that searches for communities in large networks, is all about enhancing the modularity score of each community. Modularity is a measure which evaluates how good or efficient it is to assign nodes into communities. This means evaluating the extent of connections between nodes within a community compared to what would be expected by chance if they were randomly connected across the entire network.

Figure 6: Community detection with Louvain algorithm in Neo4j

Method of Louvain is a type of hierarchical clustering. It repeatedly merges communities into one node and conducts modularity clustering on the condensed graphs.

The Louvain algorithm is a method frequently utilized in network study to identify communities. It is designed for dividing a network into groups of nodes (communities) that possess more robust internal links compared to their connections with the larger network. The Louvain algorithm, it's a technique which used to analyze and divide networks. It works through an iterative process that enhances the modularity of the network, which assesses how well nodes are grouped into communities.

This begins with the action of assigning each node to a community, so in the start all nodes are their own independent community. The main aim of Louvain algorithm is to enhance quality function which we call as modularity through modularity optimization. Modularity is a way to measure the level of interconnections within groups compared with the connections between groups. Higher modularity means that the network has been divided into separate communities more effectively.

The Louvain method uses a strategy of greedy optimization that adds up communities one by one to find the highest modularity. The process is simple: each node is looked at

alone, and we check how moving it to another community nearby would affect modularity. If there's an increase in modularity (the change in value is positive), the node gets moved into this nearby community. This process keeps going until there is no further improvement in modularity by moving nodes.

After the initial optimisation, the method of combining nodes from identical communities into a single "super-node" is used. This stage reduces the network's size and speeds up subsequent rounds.

Iteration means the repeating process of improving modularity by moving nodes between communities and joining communities. This goes on until no more progress in modularity is possible, leading to a point of convergence where the algorithm arrives at its last community structure.

When the Louvain algorithm is stable, every node of the network gets assigned to its own community based on last division. Nodes within a same community tend to have denser connections among themselves compared to nodes in different communities.

2.4 Web Scraping

The amount of digital data accessible on the World Wide Web is now estimated in zettabytes. The immense collections of Big Web Data are now being seen as a valuable strategic asset, on par with land, money, and oil. Organisations may collect and analyse this vast amount of web data to acquire a comprehensive knowledge of their internal and external environment, ultimately leading to improved organisational performance. Due to these prospects, the process of automated extraction and organisation of Web data, often known as Web Scraping, is becoming used in research initiatives.[14]

The process involves the creation and implementation of two software applications: a crawler and a scraper. A crawler is a tool that methodically collects data from the Internet, while a scraper is used to extract important information from the downloaded data and store it in an organised fashion.

2.4.1 Challenges and considerations

- Inaccessible websites

Another common occurrence in web scraping is when websites are not present. This might happen because the server is down, there are network problems or the site itself is undergoing a restructuring process. So it's very important to have good ways of dealing with these cases such as using strong error handling techniques that can detect and manage HTTP problems like "404 Not Found" and "503 Service Unavailable" while accessing websites. This may entail retrying queries, documenting problems for analysis, or bypassing unreachable websites.

Websites can take action against web scraping bots by banning or limiting them based on the user-agent string. This is why you sometimes observe a rotation of user-agent strings or using ones that are similar to browsers, all in an effort to evade detection and enhance accessibility to the website. Additionally, rotating proxies could help bypass IP limitations set by a website and prevent IP ban. For blockage prevention, web requests are directed through a collection of proxy servers that have different IP addresses.

- Dynamic content loading

Websites that use AJAX (Asynchronous JavaScript and XML) heavily in their designs might have content which is loaded dynamically. This means the desired content might not be directly present in the HTML source code of a page but rather gets loaded dynamically as

users interact with it on website. Web scraping technologies of usual manner, that is just getting unchanging HTML, will miss out on this material loaded dynamically.

Many websites employ frameworks like React, Angular, or Vue.js to make client-side rendering easier. The server sends a short HTML page along with JavaScript files. When these scripts get run by the browser, it displays all information. For scrapers, this presents a difficulty as they need to read and execute JavaScript just like a browser does for accessing complete information.

One way to work around this problem is by utilizing a headless browser, such as Puppeteer or Selenium. The term "headless" means it's a web browser that operates without any graphical user interface (GUI). These types of web browsers can execute JavaScript code like regular ones do. In this manner, they have the ability to extract dynamic material too.

A number of elegant web scraping technologies are designed particularly for JavaScript rendering. They possess the capacity to hold on until AJAX queries and the page reaches a desired state before extracting information.

In this project, we use Beautiful Soup for web scraping because it can handle many HTML parsing techniques. It is flexible to process unprocessed HTML strings, parse files or even parse HTML that has been obtained from web sites through HTTP requests. This adaptability makes it easy to incorporate into different web scraping processes.

Beautiful Soup works well with many Python modules and frameworks commonly found in big data applications. It can be seamlessly applied alongside data processing and analysis instruments such as Pandas, NumPy, and Scikit-learn. This feature makes it more convenient to manipulate and analyze data in an efficient way.

Beautiful Soup is purposely designed to be simple and easy for users. The software has a direct API that is easy to understand, along with Pythonic syntax. This makes it user-friendly for people who are at different levels of expertise in using such tools - from beginners all the way up to experts. It being so straight forward helps reduce how long it takes for development, leading to better effectiveness when dealing with big data projects.

Most importantly, we aimed to pull out hyperlinks from webpages without getting into complex intricacies or setting up inner website situations.

- Implementation of IP Bans and Rate Limits

Websites apply IP bans and rate limitations to manage the flow of incoming traffic and protect their data. There are different motives for websites to put restrictions, which are:

o Protect Server Resources: Frequent and aggressive scraping actions can overload a server, affecting the normal use of the website for regular users.
o Protection of data: This means the actions that websites do for keeping their data safe from being taken out and used by competitors or for unauthorized activities.
o Security Protocols: When there is a large amount of network activity coming from one IP address, it could be viewed as a possible security threat. For example, if a website is under attack from Distributed Denial of Service (DDoS), it might result in that site blocking access for the specific IP address.

These difficulties, while they can cause problems, have solutions to avoid them:
o Having different IP addresses in a range could help to not get caught. This can be done by using proxies or VPN services.
o Following Rate Restrictions: It is very important to understand and follow the rate restrictions set by a website. This means distributing requests equally and keeping scrape frequency within acceptable limits.
o Using random request intervals can make the scraping appear more like human behavior and lessen the chances of it being discovered by scraping at different times, instead of regular intervals.

- Legal and Ethical considerations

Web scraping, though a powerful tool to collect data, has some legal hurdles especially with private and copyrighted materials. Understanding these legal matters is crucial for every firm or person doing online scraping.

Copyright infringement is a significant legal concern when it comes to online scraping.

Scraping is obviously not allowed according to the Terms of Service (ToS) of numerous websites. Doing actions that are against these rules might be considered as breaking the contract. When talking about legal and moral issues, it's very important to think about these things such as using copyrighted information without permission, especially for making money, can bring about legal disputes. It might mean getting a message to stop and desist or being taken to court with possible punishments.

Even though the capacity to enforce can vary, there have been cases where legal actions were initiated against scrapers for violating Terms of Service (ToS), leading to penalties or prohibitions.

- Data in Large Volume

One of the main challenges in most cases is handling the big amounts of data that are usually gathered. This problem concentrates on making sure there's an effective way to handle and process all this information.

o Volume: Web scraping can generate large amounts of data that may be difficult to store and handle.

o Variety: The data comes in various formats, such as text, numbers, pictures, descriptions and reviews. This makes the processing more complex.

o Velocity: In a system where you need to do scraping again and again over certain period of time, there is a requirement for a system which can handle frequent updates and changes in data efficiently. But in our project, it is not necessary.

Here are some ways that may be used to achieve efficient data management.

o Storage Solutions that can be Scaled Up: We must employ cloud storage or distributed databases, which can grow as our data increases in size.

o Data Processing and Analysis Tools: Using trustworthy tools and platforms to process and analyze the data. Advantages can be found in SQL databases, NoSQL databases as well as big data processing frameworks like Apache Spark.

o Regular data cleaning is a methodical process that involves frequently cleaning and assessing the data to keep its quality and suitability intact.

Furthermore, when we scrape, we may encounter the different definite links that lead us to the downloadable big files, compressed files which results in the unwanted outcome in our situation. This is not what we desire for our project - these links should send us to a separate webpage. To prevent this type of cases from happening, it is necessary to examine content-length header prior to taking any action.

- Security Measures

One of the most difficult obstacles is successfully manoeuvring past security barriers such as CAPTCHAs and other anti-scraping technology. The purpose of these security measures is to safeguard websites from automated access, therefore increasing the complexity of scraping attempts.

- Scraping activities may be disrupted when a CAPTCHA is encountered or when an anti-scraping mechanism is triggered, causing the scraping operation to come to a stop.
- Heightened intricacy: Successfully navigating these security measures frequently necessitates the implementation of more advanced and sometimes more resource-intensive solutions.

Methods to alleviate security obstacles:

- Utilising a collection of rotating proxies may be advantageous for circumventing restrictions on the number of requests and prohibitions based on IP addresses.
- Headless Browser Configuration: Configuring headless browsers to accurately simulate human behaviour might aid in evading discovery.

- Integration of data and its usability

One of the key difficulties in web scraping is not only gathering data, but also efficiently integrating and making it practical for business objectives. This stage is essential for transforming unprocessed scraped data into practical and valuable insights.

- Data obtained from multiple websites may be presented in inconsistent forms, necessitating the need for standardisation.
- Concerns about the quality of data: It is essential to guarantee the precision and dependability of integrated data, since mistakes might result in incorrect business choices.

Effective Integration Strategies:

- Data Cleaning and Transformation: Establishing resilient procedures for purifying, verifying, and converting data into a uniform structure.
- Using Integration Tools: Employing data integration tools and platforms that help automate and simplify the process of merging data from many sources.[15]

2.4.2 Implementation

The implantation of web scraping in our project demonstrates a detailed and organized method for gathering and examining data. From using caching methods, parsing URLs and checking their validation to saving HTML content, cleaning link algorithms as well as extracting metadata - these show that the system is designed to be efficient, precise and useful with collected information. This strong structure not only aids in pulling out important understanding from the internet but also sets up base for complicated study of data and analysis. It promotes new ideas and findings within information science area via technology advancements.

- Cache mechanism.

Caching is a very important part of the web scraping process. It helps to store data that was previously taken, so there are less repeated requests made to web servers and overall efficiency gets better. System does this by changing scraped data into a JSON file, then reverting it back again when required - this method decreases need for repeating HTTP requests which can use up resources and take time. This also enhances the fault tolerance of our system significantly.

- Parsing and validation

Parsing and validating URL are very important parts of web scraping. They help make sure that the data collected is complete and reliable. Parsing a domain name from an URL is a process where we take out the main part of the URL, which tells the computer where to find information on internet. It's important for this step to confirm if parsed text is actually a valid domain name according to standard rules set for URLs. By using known parsing libraries and techniques for validation, the system makes certain only correct URLs get processed correctly; this lessens chances of mistakes in data and other problems like that.

- Content managing

The task of saving HTML content locally has two main reasons: preservation and making it available. When we save the HTML content from web pages into our own files on a system, it helps to keep raw data in its original condition for later study or reference.

Moreover, storing this content locally allows one to access and use the information offline; we can continue researching and examining it even when not connected with internet services. This implementation ensures that website content is maintained throughout the latter phases of development, allowing for the exploration of content analysis or the use of machine learning algorithms if necessary. Initially, the scraper does not engage in content analysis.

- Data preprocessing

Extracted reference URLs might be found in several unwanted forms. Consequently, the data required to undergo certain cleansing operations. On some websites, we came across with the situations where the data given inside the link tags includes phone numbers, empty values, or subpages of the same domain, among other possibilities. Therefore, certain utility methods must be implemented to eliminate duplicate and incorrect URLs from a list of links obtained from a webpage. The programme also use a set data structure to keep track of visited URLs and eliminates duplicate and invalid URLs.

- Meta Data Scraping

The major part that web scraping depends on is meta scraping which basically serves the role of quickly filling up the details that are located within the internet pages. The mentioned metadata, of course, will comprise several context-released compounds that include titles, description, keyword, as server/location data too. It is particularly useful when you want to understand what is on webpages and how everything is structured. The method we practice appraising the text of HTML is a systematic one. The tags we gather are meta/meatags and there is a great need for better informations and analysis of this data.

Meta scraping impacts considerations in the areas of information acquisition, content analysis and search engine optimization. As a consequence of our technology's dynamics, researchers, analysts and webmasters are able to collect all the important metadata elements that help them to draw a complete picture of a material's properties in the future. The contextual details allow the pages to be placed in the website's correct context and space. Hence, they help in categorizing and organizing web pages, based on the web site's circumstances.

Scraping in a disciplined way is an activity of looking at webpage source code. HTML is example of such sources. By analyzing crucial meta tags, it finds out the source traits, builds up the dataset and finally it answers the query. With our algorithm moving swiftly across the same hierarchy of web pages and picking particular parsing libraries and algorithms, systemic work is achieved. It helps us in properly placing the needle on that specific line and as a result, it is very accurate in identifying objects in certain areas. The title, description and keywords take the important metadata part and each of them is read with due caution to ensure their accuracy by careful looking to catch possible patterns. The approach upholds the originality and reliability of the data one gets.

Meta scraping serves a variety of purposes, including collecting material from different origins, analyzing competition and improving semantics. It helps in the making of big datasets because it can gather details from many sources. These sets are useful for studying trends or markets and making business intelligence better. Also, meta scraping provides a way for website owners to monitor and understand the metadata of their particular web pages. They can use this information in order to improve these pages which helps increase their visibility on search engines as well as interaction with users.

Continuing to the previous point, as online technology keeps advancing, meta scraping presents advantages and challenges for scholars and workers alike. Possible future progress in natural language processing, machine learning, and semantic analysis could bring about new capabilities to extract and comprehend metadata. The continuous requirement of studying and inventing in the field of online scraping is demonstrated by problems like data quality, scalability along with moral concerns.

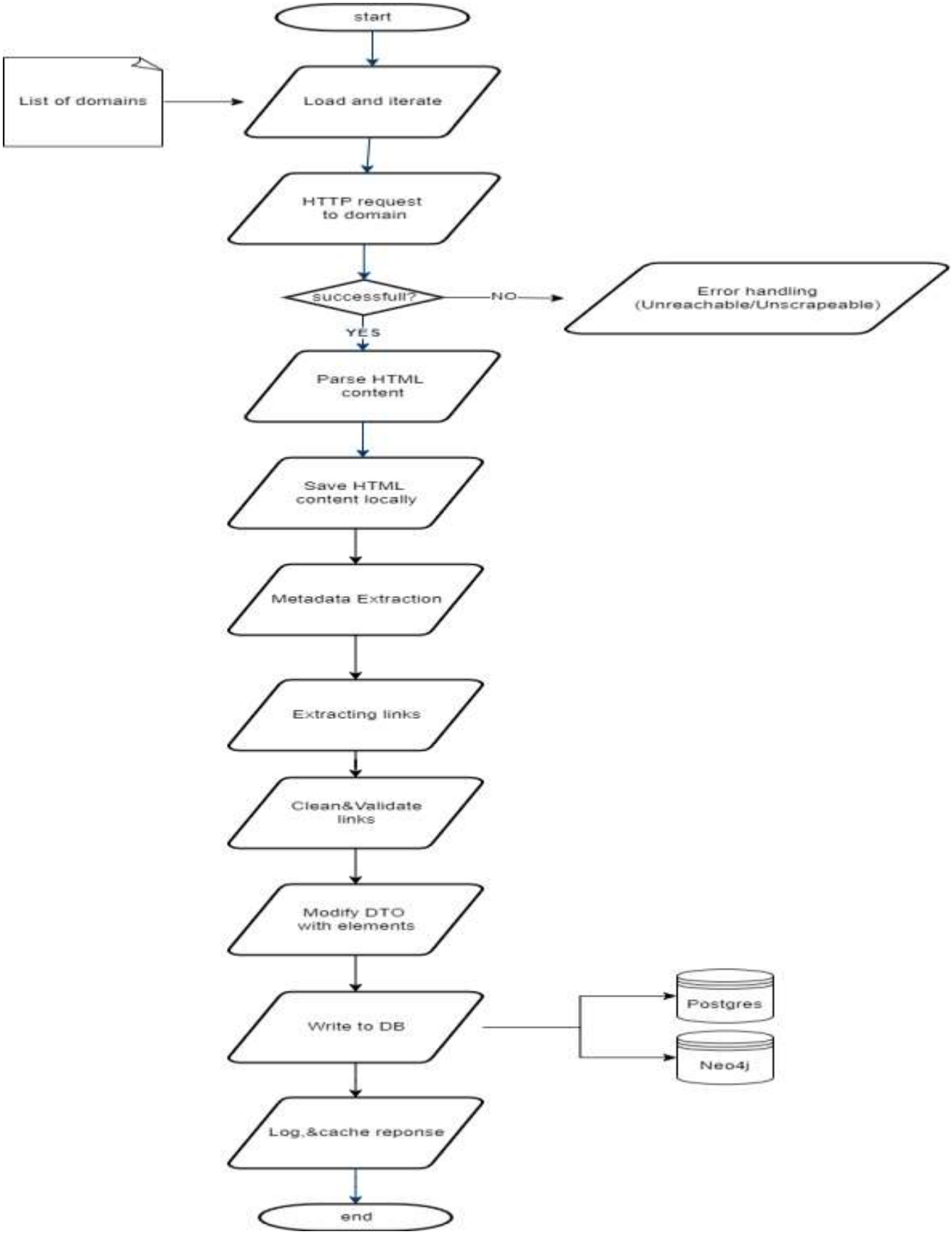

Figure 7: Workflow diagram of Scraper

2.5 Visualization

Many researches have been done about big data visualisation. Before picking a technology for visualisation, it is important to measure our data, its traits, the visualisation technologies we can use and any difficulties that could come up in making progress.

2.5.1 Issues

The process of creating visualisations for graphs deals with several hurdles and worries that are natural to graph visualisation. These concerns can include:

- Scalability: Handling big and complex graphs might present problems, for example in rendering speed and interaction with the user.
- Layout: Finding the best way of organize nodes in a graph, so they show interactions between each other and not get mixed up or cluttered together.
- Interactivity: This means the graph has interactions which let users to explore and modify it. These can be functions such as zooming, moving around or filtering.
- Visual Encoding: This term means picking the right visual aspects like color, size, shape etc., for showing parts and characteristics of a graph clearly understandable to viewers.
- User Experience: Making the visualisation easy to use and understand, providing useful knowledge without overloading with too much detail.
- Compatibility: This refers to the software or website functioning properly on different devices and browsers. It includes elements like how well it adjusts to various screen sizes and how efficiently it runs.

Dealing with these problems requires an in-depth study of design choices and the use of correct tools and platforms for graph visualisation.

2.5.2 Complexity:

The complexity of graph data arises from the complex interconnections between nodes and edges and this complexity derives from:

- Graph Size: The size of a graph may vary significantly, including a broad range of nodes and edges, from tiny, uncomplicated graphs to vast, highly interconnected networks.
- Structure: The complexity and visual depiction of a graph may be influenced by its structure, which may include hierarchical, directed, or cyclic interactions.
- Attributes: Data properties in graphs might include labels, weights, and metadata, which contribute to the intricacy of visualisation.
- Dynamic graphs: Which depict the changing nature of graphs over time, provide further intricacy since they need animation and temporal analytic capabilities.

To effectively visualise complicated graph data, it is necessary to use approaches that simplify, abstract, and portray the data in a meaningful way. These techniques aid in understanding and analyzing the data.

2.5.3 Software tools and technology platforms:

Visualization tools and systems vary in regard to architectural (or design) features and capabilities. Commonly used tools and platforms are:

• Being a bunch of the tools such as D3.js JavaScript language makes the data-visualizations like graphs or networks to be more flexible and adjustable.

• Neo4j Bloom: It is a kind of tool that has a feature of graph visualization specifically for Neo4j databases. It brings more convenience in studying and using the graphs in class is fairly simple to grasp.

• Gephi is the name for an open-source and also free networking program for visualizing and analyzing links. It equips you with sophisticated algorithms to arrange network items, and offers interactive features for visualization of data in a way that will make it also fun and exciting.

• The data visualisation and network analysis program called Cytoscape is set up and complete with the necessary modules to run different analytical applications in graphs.

Depending on the essential goals and needs, the choice of tools and platforms will rely on the requirements, programming skills level, and of course the required functionalities for graph visualization.

2.5.4 Graph Layout Algorithms

A basic way to display this type of data is by using node-link diagrams. These show the connections between the pieces of data with lines. A different method suggested for visualizing graph structures is displaying them using space-filling methods or space-nested layouts which show relationships in an implicit manner. The first need for node-link architecture is about calculating the nodes' coordinates and drawing the lines. To make it more readable, a clean layout needs to fulfill these requirements:

- The nodes and edges should be distributed evenly.
- Strive to lessen the amount of edge crossings.
- Representing symmetrical subgraphs uniformly
- Reducing the edge bending ratio
- Reducing the lengths of the edges, which helps readers quickly identify the connections between distinct nodes.
- If the data has a structure that is easy to understand, it should be divided into layers. This makes the graph underneath more understandable. For example, when working with data-flow diagrams, it's recommended to categorize graph pieces into multiple layers so we can guarantee the final depiction matches what's being represented at its core.

In reality, it is difficult to put together most of these requirements. Not all of them are in harmony. In practice, most algorithms are a mix or compromise as they balance different aspects and make trade-offs to achieve the desired outcome. Choosing what criteria is needed for an application is not a strict procedure but changes based on the situation. Setting up an order for criteria is an important step before choosing layout algorithms that fit well with this hierarchy of needs.

Several techniques may be used to determine the layout of a graph, including the topological Feature-Based Layout, the Spring Layout Algorithm, and Node-Link Tree layout

techniques, among others. This project will mostly concentrate on the Spring Layout Algorithm.

The method of spring layout, which is also known as force-directed layout and was first presented by Eades in 1984 [18], represents a type of node-link structure. The structure created through the use of force-directed process is frequently seen as a popular node-link configuration due to its simplicity and ability to provide symmetrical arrangement. The approach for spring layout involves modeling the network as a physical system. In this system, the graph nodes are considered charged particles that are connected together by springs forming an array. Each node is connected to two types of forces: attractive forces and repulsive forces. This method attempts to minimize the total energy of the spring system by adjusting node positions, using provided node coordinates and spring characteristics. The force of attraction, fa, is applied to the connected nodes that are linked by a spring. On the other hand, the force of repulsion is applied to all nodes in the network. Here, we give these forces some definitions:

$$f_a(d) = k_a \, log(d)$$

$$f_r(d) = \frac{k_r}{d^2}$$

In the formula, ka and kr are fixed values. d is the distance between two nodes at that moment. For example, Figure 8 shows a short example made with this method. Even though the forcedirected technique makes graph layouts look nice and even for medium-sized graphs, many calculations are needed which makes it one of the expensive algorithms computationally. Algorithm's temporal complexity is more than O(n^3), where n denotes total nodes. Moreover, in respect to predictability, force-directed layouts are lacking as running the algorithm twice produces varying results. This makes it difficult for unstable layouts to maintain the user's cognitive representation during interaction with such a layout. Though there are a few shortcomings with the force-directed layout technique, it has been widely employed in many visualisation frameworks. [17]

Moreover, the technique has undergone several revisions and optimisations to address its inherent limitations.

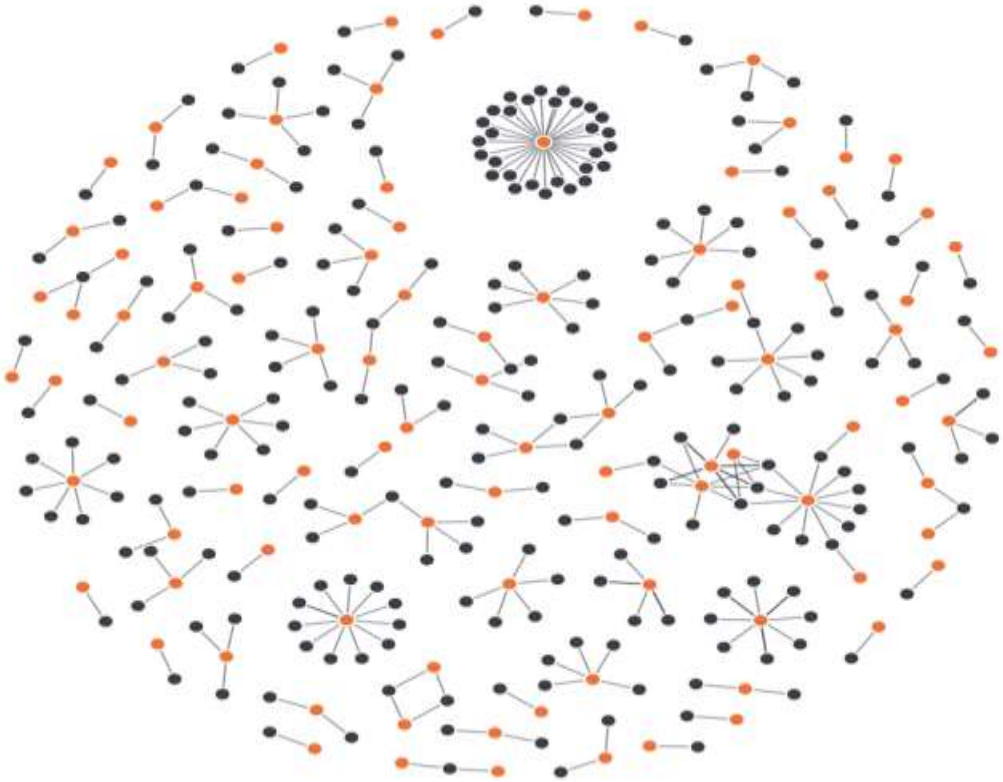

Figure 8: Force Directed Layout sample

2.5.5 Clustering and Community detection

In the graph visualisation, clustering methods were employed to handle visual congestion. These techniques assist in producing conceptual representation for the initial graph. The elimination of numerous visual components aids in both clarity and comprehension of the final arrangement as well as improves rendering efficiency.[17]

Algorithms for clustering can be separated into two main groups, which depends on the criteria used in clustering process. The first method is called natural clustering where structural information among nodes within network helps to find a pattern of nodes having alike criteria. The next clustering method is content-based clustering. It looks at the semantic meaning of relationships between nodes in the network. This type of clustering isn't used much because it relies heavily on application domain, making it not practical to reuse the same content-based clustering method in another application. So, most of the graph visualisation tools employ clustering methods that are based on structure. Different structural properties have been utilised as grouping criteria like distances between nodes in a graph and degrees of node. Natural clustering technique is frequently used to preserve the original network's structure integrity. This type of clustering can improve interaction abilities because

it makes filter procedures easier to apply on the layout result, leading to faster search speed for some data patterns. We could do this by dividing nodes into separate groups, applying a filter according to certain criteria and finally narrowing down the search area only in surviving clusters.

For this project, we used community detection to cluster nodes and so streamline the graph, reducing loading time and other related factors. To do this, we use the Louvain method, which is included within the graph analytics package of neo4j.

Essentially, we assigned a community attribute to each node to indicate which community they are part of. This allowed us to visualise the whole graph and also the graph of communities when the user zoomed out.

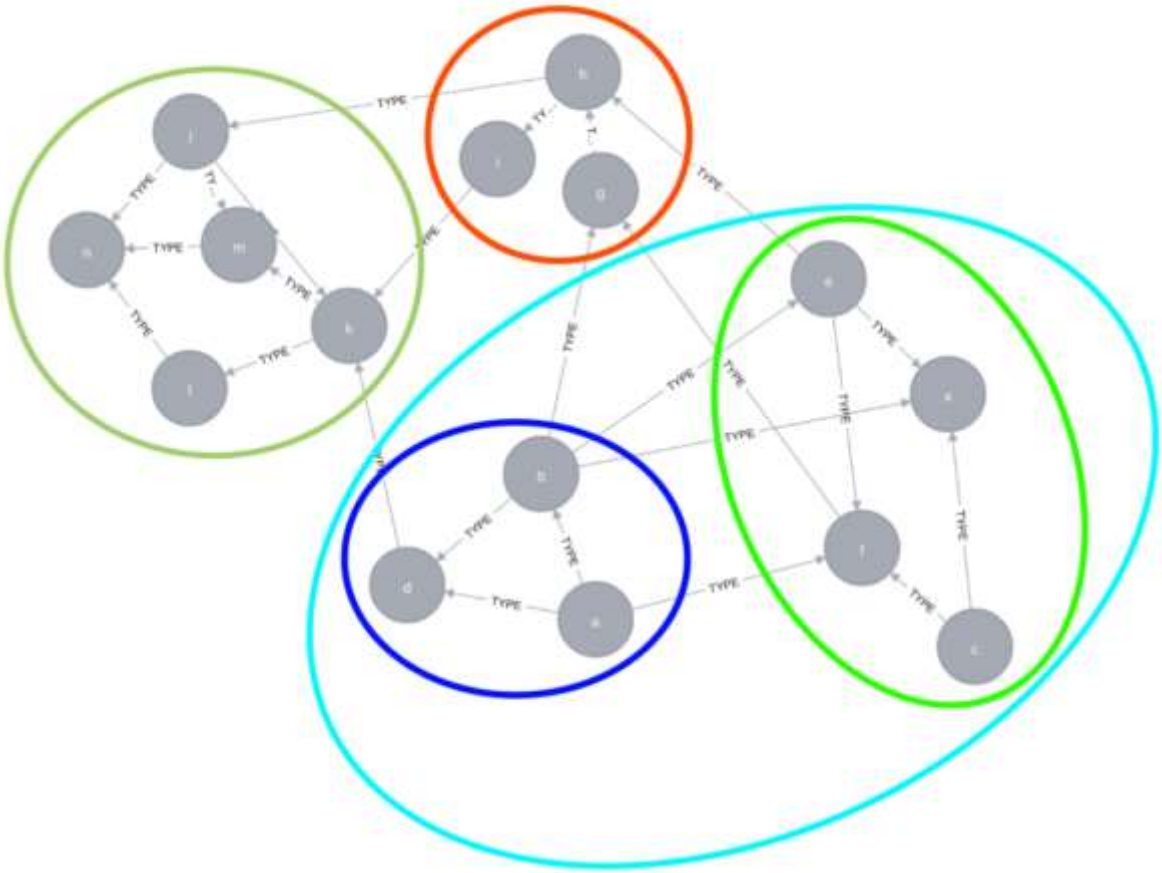

Figure 9: Community detection

2.5.6 Interaction Techniques

The main goal of visualisation methods is to improve the comprehension level for given data. Many times, this aim cannot be achieved merely by creating a static image that represents the data. We need to provide an ability for interacting with information. Therefore, a lot of studies have been done to create better methods for interface and navigation. These methods aim at improving how efficient data exploration tasks are completed.

Zooming and panning are fundamental techniques for navigating through large volumes of data. Panning refers to the movement of the camera over the scene, while zooming enables viewers to transition between abstract and detailed perspectives. Geometric zooming modifies the screen transformation, enabling the user to either increase or decrease the magnification of the shown graph. Semantic zooming refers to the ability for both the size of objects and the displayed information to change while approaching a certain location of the graph[17] . By using the community detection methods outlined in section 2.4.5, we successfully integrated zooming actions with community detection approaches to enhance the dynamism of the web-graph.

2.6 Search Engine Optimization

To get a website famous and have high visibility on the Google search engine, it is not easy. The algorithm of Google uses nearly 200 parameters for producing search result pages (Srp). Laurea University of Applied Science investigates about 8 factors for on-page optimisation and 4 factors for off-page optimisation. The study ends with the idea that SEO raises online exposure in search engines. Additionally, the university creates handbooks that include latest SEO tactics and strategies (Deepak, B., 2017). The research [21] uses a method of gathering from internet 200 factors (Dean, B., 2016), checking their importance, making a to-do list, applying it on recently made website and observing pagerank progress in Google comparing to other websites without using SEO. Subsequently, the results are analysed.

SEO was used in this project to improve the appearance and position of Azerbaijani web pages that were collected by crawling and stored inside databases. We developed an algorithm to calculate SEO score for each web-page.

There are 6 factors affecting our SEO score evaluation. First, we consider the keywords existing in the content of the webpages. Optimally, this must identify main and secondary

keywords, assessing how much they match with webpage's content and making sure keyword density is appropriate, but since we are working with the thousands of webpages it will take too much time and computation power to dive into content for comparisons. So, we evaluate based on the existence of this parameter. Same evaluation goes for On-Page SEO Analysis as well, it includes

- Title Tags - Checking if each webpage has unique and descriptive title tags.
- Meta Descriptions: Evaluating the presence and quality of meta descriptions.
- Heading Tags: Verifying correct utilization of heading tags, like H1, H2 etc., and their relation to the content.
- Image Optimization: Checking for descriptive filenames, alt text, and size optimization of images.

Another standard to think about is Social Media Integration. We are looking at the count of social media links found on the main page. This study examines how effective it is to include social media in a website for boosting its traffic and improving search engine optimization (SEO).

During the process of web scraping, the links that exist within a web page and those that are outside it were taken out and saved as JSON data in the "links" column. Later on, these gathered links were subjected to analysis using an assessment algorithm. These facts hold importance in assessing how well-structured and influential a website's link network is - aspects highly relevant for SEO.

Apart from social media, we can also evaluate webpages by looking at broken links. Broken links mean the connection doesn't exist anymore or it's in a wrong structure.

Additionally, we check for the existence of certain metatags to see if the site is indexable. If these tags are missing, it can negatively affect the SEO score.

To sum up, the project planned to use these SEO methods during the data collection and storage stage. This was done with the goal of creating a substantial dataset for more analysis and visualization, so that we can understand better about SEO performance and possibilities in Azerbaijani websites.

3 Results and analysis

3.1 Web Scraping Process

The web scraping process is conducted by running a program that automatically extracts data from websites. During crawling, the program follows links on each page to reach other pages within website boundaries. The process continues until there are no more new pages found.

The crawling process starts with one initial URL called "seed", then it moves to follow all internal links present in subsequent webpages discovered during crawling operation - this means that our program only focuses on gathering data from those areas reachable via clicking within the site itself. Crawling stops when no new URLs can be explored or maximum allowed depth has been reached; these limitations were not set up so theoretically our program could keep going forever if there was always another page available for exploration! In total, 13447 webpages were crawled during this operation taking around 15 hours' time duration as per statistics provided by Python crawler library used here.

Stats collected at end of crawl include:

Total number of nodes: 19320

Total number of edges: 41743

Number of URLs requested but not yielded any response yet: 4462

Number of times URL redirection occurred while crawling: 1941

Number of times URL permanent redirection occurred while crawling: 700

Maximum depth that crawler reached while exploring webpages was: 1

Seed urls with HTTPS protocol: 7677

Seed urls with HTTP protocol: 1308

3.1.1 Web Scraping and Data Extraction

The process of web scraping was very important in making the graph representation for the Azerbaijani web ecosystem. We made a special Python scraper to go through and take out data from a group of websites that belonged to Azerbaijan domain, this acted as our first dataset. The scraper used Beautiful Soup library which is an effective tool for web scraping and understanding HTML contents. It helped to move around within websites and collect appropriate details from them with ease.

The main objective of the web scraping was to collect hyperlinks that exist on every website, because these links would become the edges in graph. But for a more complete view on Azerbaijani web environment, we also gathered other metadata:

1) Page Titles: The names of the web pages, they could give an idea about what these websites are all about.

2) Meta Descriptions: These are the short summaries, typically present in a web page's metadata that tell you about the content on this specific webpage.

3) Keywords: The keywords linked to every web page, that could assist in recognizing the subjects and themes discussed on the site.

4) Server Information: Details regarding the server that holds the website, such as server software, IP address and geographical location of the server.

This method of taking out metadata, when done with the hyperlinks, enabled a more complete examination and comprehension of the web ecosystem in Azerbaijan. It was possible to analyze content themes, categorize websites and explore potential reliance on outside resources.

In the procedure of web scraping, we met many difficulties. The most important one was managing dynamic content loaded with JavaScript. Regular methods for web scraping might not capture this content because it changes after loading into the browser. So, to solve this issue we might set up the scraper to use headless browsing methods with tools such as Puppeteer or Selenium. This made sure that all elements on page - including those created by active scripts - were displayed correctly when extracting data from them.

The next difficulty was dealing with IP bans and rate limits set up by websites to control incoming traffic and safeguard their data. The scraper might overcame this problem by using rotating proxies, plus random request intervals that imitated human-like actions to prevent being recognized as a scraping bot.

However, last 2 methods are not implemented in this stage of development due to reasons like time and computation power.

In the end, the internet scraping process effectively went through and got data from 13447 websites in Azerbaijani area. The whole thing took around 15 hours, during this time we gathered about 19320. This information became base for making the graph model and following study on web system of Azerbaijan.

3.2 Storage- Neo4j vs Postgres

In this study, the databases used were both PostgreSQL and Neo4j. But for different requirements, they showed varying suitability and performance. PostgreSQL is a classic relational database management system (RDBMS). It handles structured data very well, ensuring data integrity with help from constraints like primary keys, foreign keys and unique constraints. It has strong SQL support which makes it good for fast querying and managing data; this makes it suitable mostly for tasks related to tabular data analysis or reporting. It was extremely beneficial for later analysis related to the content such as SEO, GeoLocation extraction and so. However, PostgreSQL's strict schema and absence of built-in graph data handling made it harder to work with complicated graph forms and traversal inquiries. Moreover, since we are not capable of saving it as a graph, we divided graph to source nodes and target nodes which brings extra difficulty to regroup them as a graph in visualization.

On the opposite side, Neo4j showed its power as a graph database in how well it could handle representing and analyzing complex relationships found within the Azerbaijani web ecosystem. Its focus on graphs for data modeling coupled with the Cypher query language allowed easy exploration through interconnected web pages by conducting traversals efficiently. This helped in gaining understanding about network topology, centrality measures and community structures of certain parts of this large-scale linked environment which is vital for comprehending how information spreads across different sections within it. Neo4j gave great speed in dealing with graph-related tasks, but it didn't

have the strong data integrity features and experience of PostgreSQL for handling table-like data.

The decision between these two databases relied on the particular analytical needs. PostgreSQL was very important for keeping and studying structured data, while Neo4j was exceptional in operations that are focused on graphs as well as graphical presentation; this allowed a complete comprehension of Azerbaijan's web environment.

3.3 Visualization Techniques

3.3.1 Techniques and Reasons

The method for visualizing the Azerbaijani web graph was implemented with D3.js, a strong JavaScript library used to make interactive data visualizations in web browsers. The visualization includes a few significant techniques:

- Force-Directed Layout: The graph's nodes and edges were placed using a force-directed layout algorithm. This method imitates physical forces between the nodes, producing an attractive arrangement that emphasizes structural patterns and connections within the graph.

  The force-directed layout helps to show a clear picture of the web graph structure. Nodes are positioned according to their connection strength, where nodes that link closely are put nearer together. This layout makes it easier to find areas with dense connections and central nodes in the graph. We can see fisheye view of our graph and

the layout on the (Figure 10). The image is just to show the distribution of nodes in the graph.

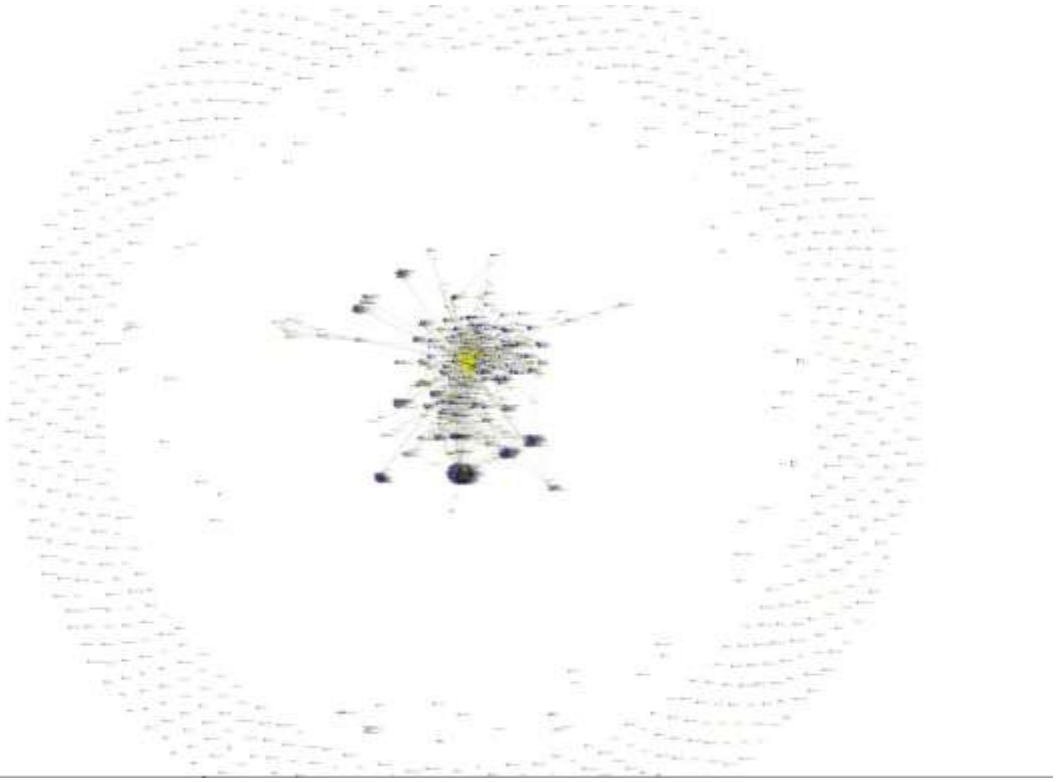

Figure 10: Fish-eye view of graph.

- Community Detection: The visualization uses community detection to find clusters or communities in the web graph. Nodes from a single community are shown together, allowing users to understand compact substructures inside the big network.

  Community detection makes the web graph more understandable. It shows hidden patterns of structure and unity in the graph. Users can see clear communities or sets of websites, which are parts that stick together within the whole network. This function lets people explore groups with similar themes or interest areas within this system called "web". (Figure 11)

- Zooming and Panning: For giving users the ability to interactively zoom in or out, as well as move around on the graph by panning. Users can investigate various levels of detail within the graph by zooming in or out. They may focus on particular areas of interest by zooming in and explore them more thoroughly. On the other hand, they can get a wider view of overall graph structure if they choose to zoom out. The

panning feature lets users navigate through different parts of their graphs; for example, they might move left or right or go up and down (move across) like moving around a map's landscape with their mouse pointer while holding down an appropriate button on it for continuous navigation action without needing any additional clicks each time).

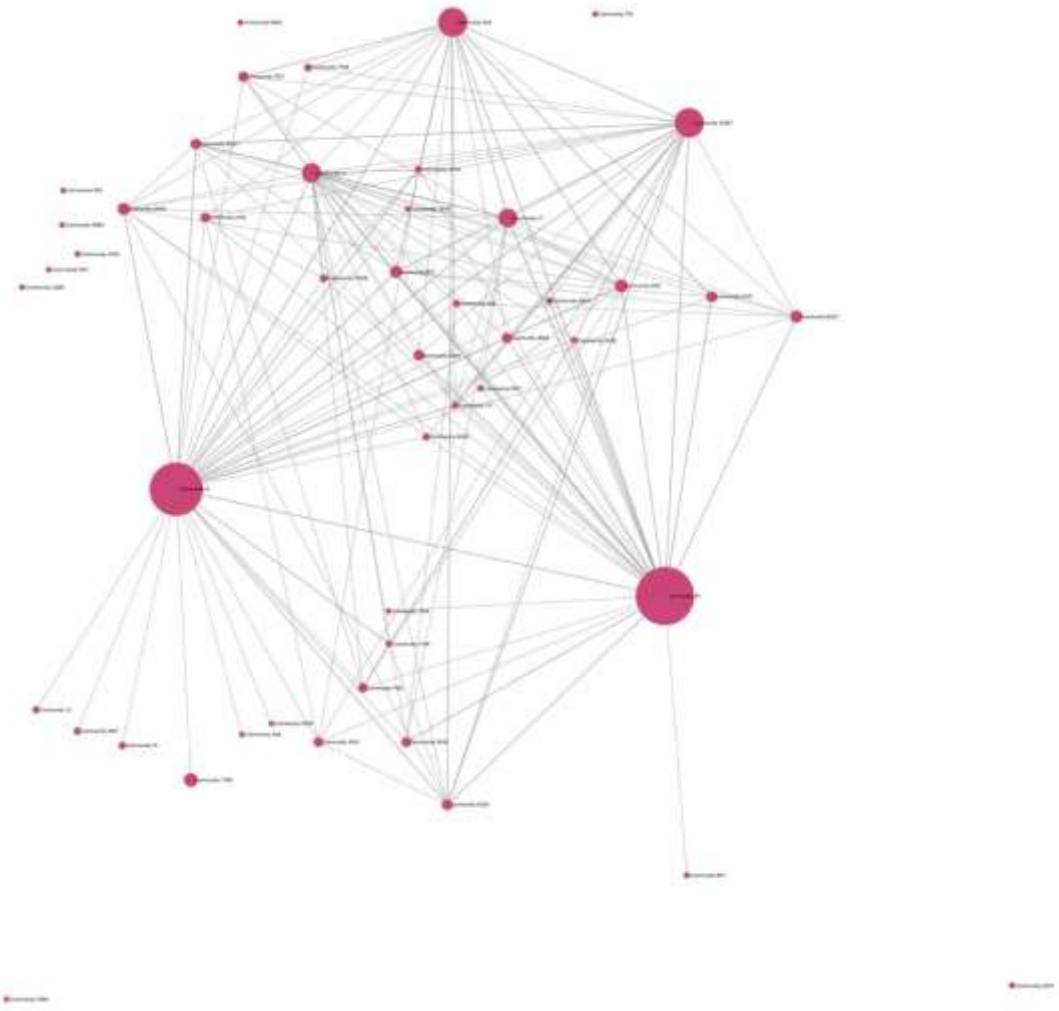

Figure 11: Community Graph

- Filtering: We have added a dashboard containing filtering options based on Node properties. Users can input filter values in the search interface, which will then fetch and show the subgraph falls under that specific demands. This feature helps in focusing exploration and study of single website or sections of the web graph. As a result, we can filter and view graphes based on:

- Betweenness centrality

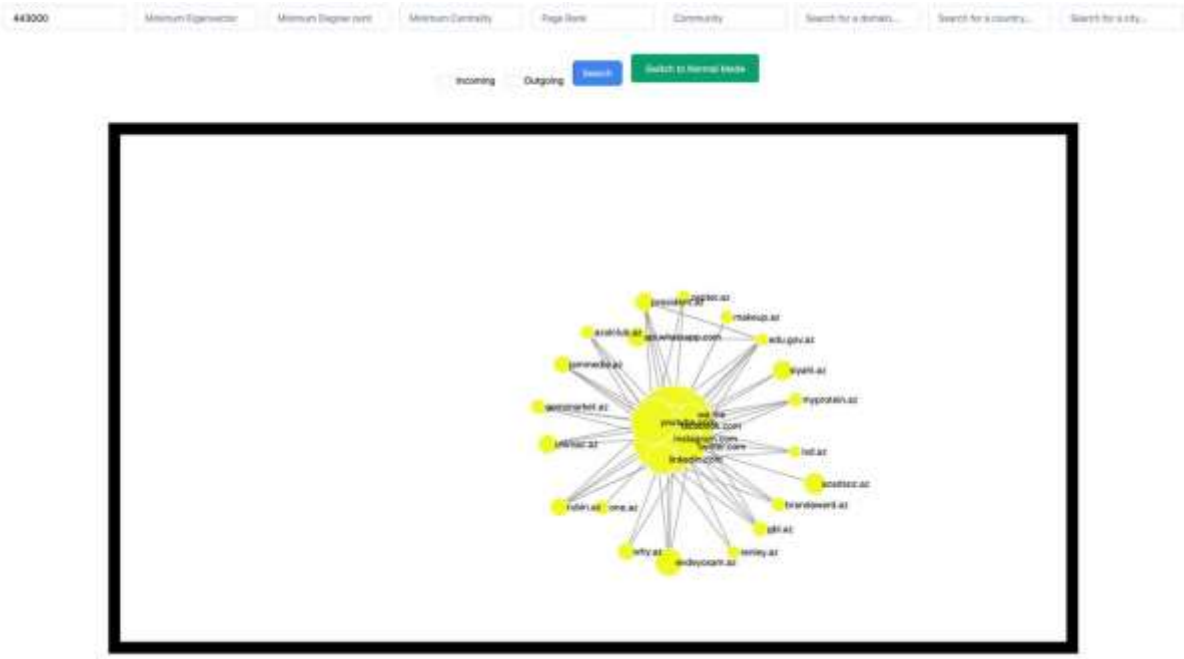

Figure 12: Filter by Betweenness Centrality

- Eigenvector centrality
- Degree centrality
- PageRank score
- ArticleRank centrality
- Community identities
- Country

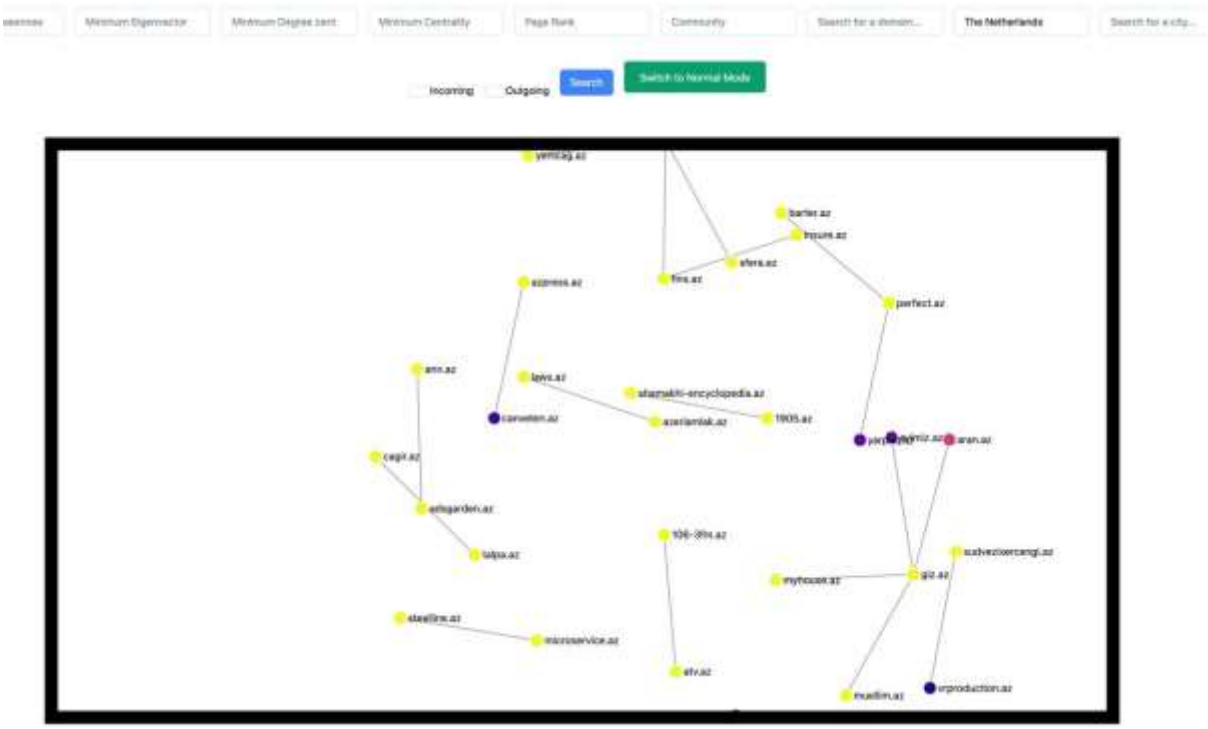

Figure 13: filter by Country

- City
- Domain name (Figure 14, 15)

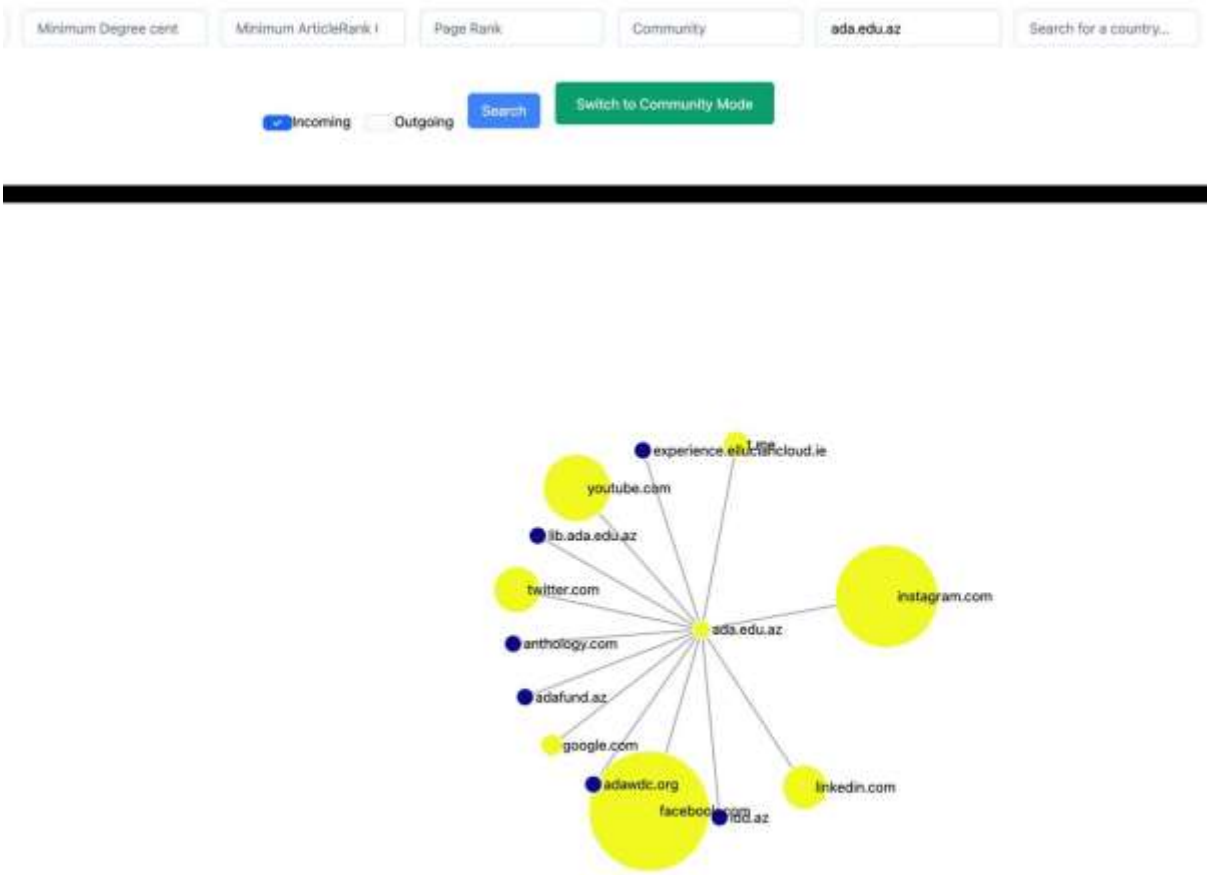

Figure 14: Webpages which are referenced from ada.edu.az

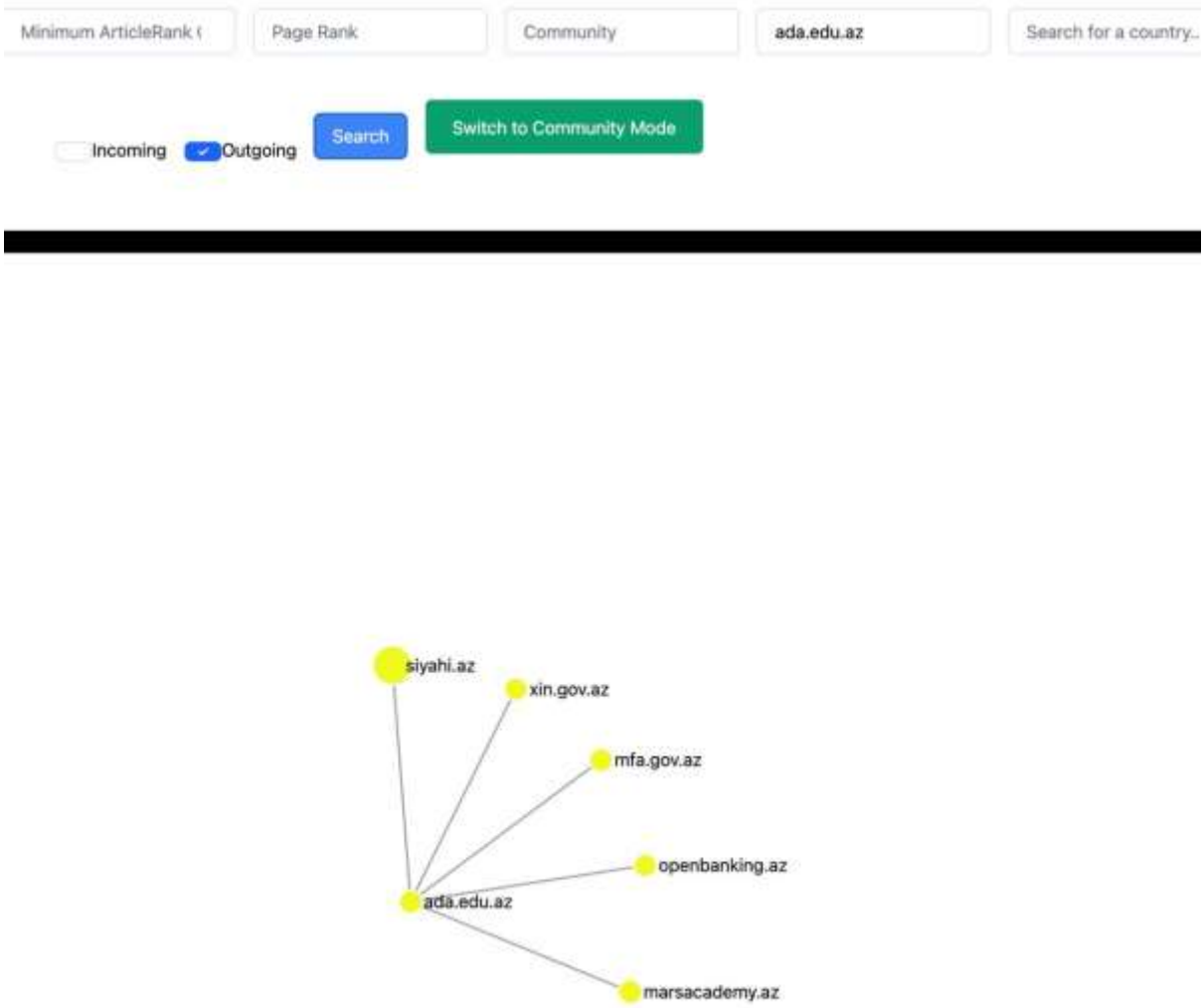

Figure 15: Webpages which are referring to ada.edu.az

3.3.2 Examples and Key Features

The visualization gives users a deep and active involvement, helping them to understand the makeup and changes within the Azerbaijani web graph. It includes several important aspects:

- Node Representation: Each node in the graph is a website, and its size and color represent extra characteristics like PageRank and betweenness centrality.
- Edge Representation: Edges, which are lines joining nodes, show hyperlinks among websites. They help to demonstrate the connectedness within web graph visually.

- Community Visualization: Websites are displayed as groups or communities, making it easy for users to recognize sets of related sites. (Figure 8)
- Interactive Node Selection: Users can interactively choose nodes to see detailed information about their properties, such as URL, page title, community membership, centrality measures and geographic location. When the graph is in the community mode, clicking on a node will display community properties such as members.

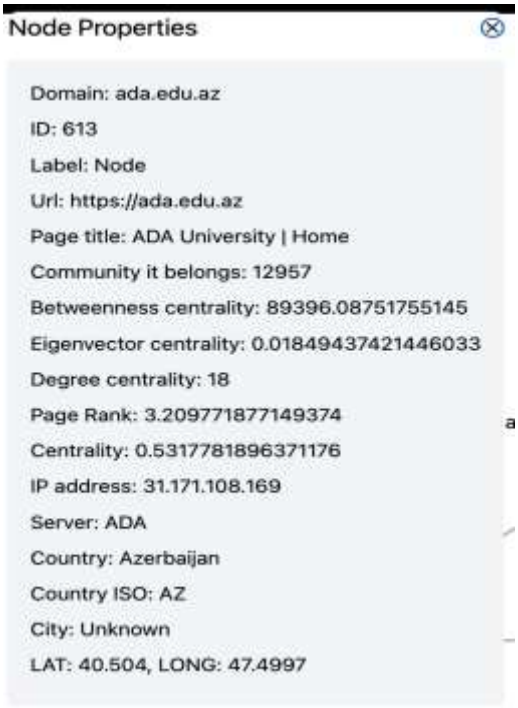

Figure 16: Node Properties

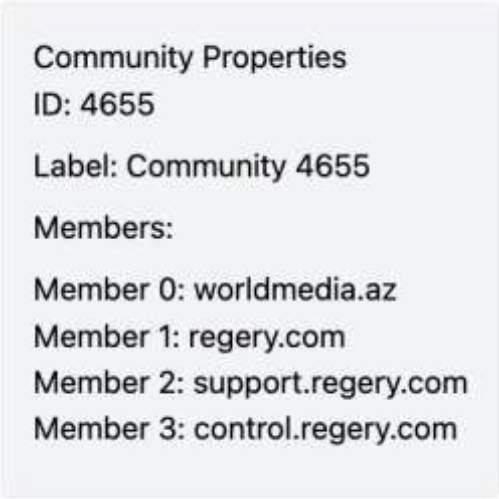

Figure 17: Community Node Properties

3.4 Page Rank

The study of PageRank scores gave us important understanding about the arrangement and importance of nodes in Azerbaijani web graph:

- Node Importance: The PageRank scores are showing how important or influential every node is in this web graph. Nodes with higher PageRank scores are seen as more influential, indicating a stronger connection and more incoming links from other major websites.
- Identifying Central Nodes: The nodes having top PageRank scores are similar to hubs or main authoritative sources in the web graph. These nodes work as crucial entry points for exploring the network and probably get lots of traffic and focus from users.
- Ranking Websites: PageRank scores permit the ranking of websites according to their significance in the web environment. Websites having greater PageRank scores are placed more prominently in search engine outcomes and seen as more reliable places for information.

Domain	Page Rank Score
Facebook.com	747.30899
Instagram.com	618.70927
Youtube.com	372.43303
Azad.az	273.34481
Twitter.com	212.55394
Linkedin.com	204.80161

Figure 18: PageRank scores

3.5 Geographic Visualization of Server Locations

The web graph was used to create a geographic visualization of server locations. This helps us understand where the web servers hosting Azerbaijani websites are located in space.

When we plot these server locations on a map, it becomes possible to see clusters of servers in certain areas or countries. This visualization also shows the worldwide extent of Azerbaijani websites and their hosting infrastructure. The following steps are followed for getting data from the web graph for creating a visual representation:

- Data Extraction: Websites' metadata, like IP addresses, were extracted using web scraping tools from Python such as BeautifulSoup. This process involved getting into web pages, making sense of HTML content and taking out data linked to servers.
- Mapping Geolocation: I utilized the GeoLite2-City.mmdb database to map IP addresses into geographic coordinates (latitude and longitude). This database gives precise geolocation details according to the IP address, which helped me find where servers are situated.
- Data Visualization: In a React application, we utilized D3.js library to create an interactive map that showed the geographic coordinates of server locations. This map gives a visual representation of how servers are spread out in different parts of the world. (Figure 12)

Figure 19: Ip distribution all over the world

We can also view the results from the below table. (Figure 16)

Country	Number of Nodes
United States	1974
The Netherlands	1470
Azerbaijan	1414
Germany	1138
Russia	767
United Kingdom	350
Finland	316

Figure 20: Top host countries

According to the results, only 17% of websites hosted in Azerbaijan which shows how much we are reliant from the outside resources.

3.6 Search Engine Optimization

We have successfully used SEO methods to gather a dataset that is useful for later analysis and visualization. This can help us understand the SEO performance and possibilities of websites from Azerbaijan. By pinpointing areas in need of enhancement and putting into action focused strategies on these sites, those who own or develop them could improve their visibility on the internet while also bringing more appropriate traffic through search engines.

The dataset contains a variety of SEO scores for webpages. The average score across all webpages is ***19.45***. This metric gives a general sense of how well the SEO is performing on these analyzed webpages.

The webpage that got the highest SEO score is ***azertaq.az*** with a score of ***37***. This shows very good optimization and sticking to SEO rules.

On the other hand, the webpage that got identified with smallest SEO score is ***alinino.az*** and it has a score of ***0***. This indicates possible areas to enhance in search engine optimization and quality of content.

Score distribution analysis of SEO scores among the analyzed webpages revealed the following breakdown: (Figure 17)

Excellent: 697 webpages got an SEO score of 30 or more, showing exceptional optimization.

Good: 1262 webpages got an SEO score from 25 to 30, showing robust optimization work.

Average: 1836 webpages showed an SEO score from 19 to 25, which is moderate optimization.

Below Average: Among the SEO scores, 3148 webpages had a score between 0 to 19 which suggests they are not performing very well and could be enhanced.

Poor: 26 webpages scored 0, highlighting significant deficiencies in optimization practices.

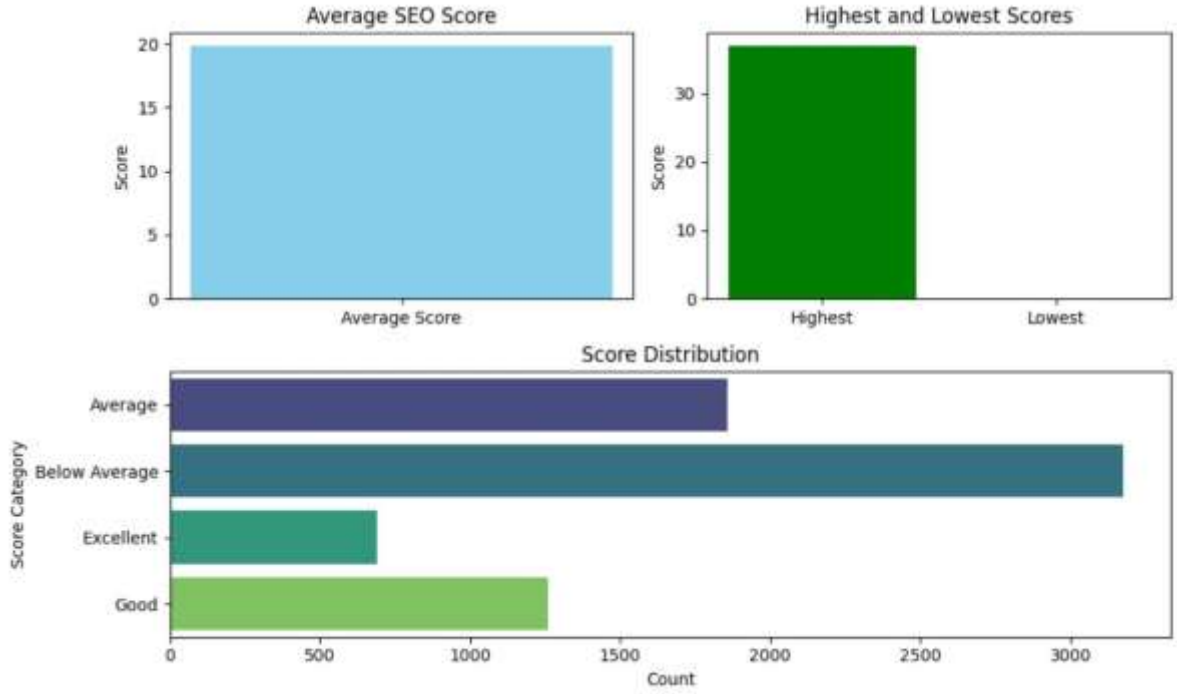

Figure 21: SEO score distribution.

4. Conclusion

Fast growth of internet and more dependence on digital technology show how crucial it is to comprehend the setup and changes in online ecosystems, especially when considering a national perspective. This study attempted contributing towards this comprehension through

performing an extensive analysis as well as graphical representation of Azerbaijani web ecosystem.

By putting into use sophisticated methods in graph analytics, web scraping and data visualization, this study has managed to make a graphic portrayal of the Azerbaijani web. This has helped illustrate its basic framework, connections within it and possible weaknesses. The results have shown us the ways things are connected, which websites hold influence and how much reliance there is on outside sources.

The study used community detection algorithms and centrality measures to find out the modular structure of Azerbaijani web. This showed groups of related websites, as well as important hubs that are central in spreading information. The examination on geographic server locations has shown us how much local sites depend on hosting infrastructure outside our country's borders - this gives a base for evaluating possible dangers and making backup plans.

The use of page ranking algorithms has made it possible to assess the importance of websites in the Azerbaijani online world. This helps in creating better tools for finding information and aids decisions about digital strategies and policies.

In general, this research has made a significant contribution to the web science area in our ecosystem. It has improved our knowledge about Azerbaijani digital environment and how it affects social growth, digital robustness, and making of important strategies. The discoveries and methods shown in this study set a base for more investigation and examination of data online. This promotes cooperation across many fields of study while encouraging methods that use data to face new problems in the digital world.

5. Future work

Even though this study has given beneficial understanding about Azerbaijan's online environment, there are many chances for future work that could enhance our knowledge and address increasing issues.

Temporal Analysis: A longitudinal research, where data is repeatedly gathered and studied over time, would let us observe alterations and interactions in the online system of

Azerbaijan. This method could make it easier to recognize growing patterns, follow how websites stay important or not as much so anymore, and notice possible shifts in depending on outside resources.

Utilization of Advanced Machine Learning Methods: The use of advanced machine learning methods like natural language processing(NLP) and sentiment analysis could enhance the scrutiny of website content and metadata. It may assist in achieving a deeper understanding about ideas, subjects, and feelings expressed on Azerbaijani internet. This might potentially reveal important viewpoints for decision makers, business people, and researchers alike.

Applying in Other National settings: Using the methodology and procedures of this work in different national or regional contexts will increase our understanding of online ecosystems worldwide. Comparisons between countries might show how websites are organized similarly or differently, leveldepths at which they rely on digital technology, and probable weak points for each nation. Such knowledge can help in forming global strategies and guidelines to improve digital strength and cyber security.

Upcoming explorations could focus on creating predictive models that utilize the findings from this study to anticipate disruptions or weaknesses within the online space of Azerbaijan. These models might incorporate various elements like website interdependencies, server placements and network structures. They can serve as useful aids in making choices related to risk evaluation and readiness for emergencies planning.

Another prospect for future research is to perform content analysis and categorization of nodes by looking at specific properties like content topics, security qualities, and other important features. Machine learning methods and natural language processing can be applied by researchers to group nodes together according to similar content topics. This aids in comprehending the structure as well as characteristics of Azerbaijani online graph more effectively.

Further inquiries need to focus on bettering the scalability and efficiency of web crawling algorithms. By taking out depth limitations and utilizing more advanced computational abilities, it becomes feasible to perform a deep study of the Azerbaijani web

graph. This will allow for a more extensive investigation into connection patterns as well as network architecture.

The investigation of moral and law elements: As online data analysis keeps growing, it's crucial to think about the ethical and legal parts tied with data privacy, intellectual property rights as well as good practices in managing data. More study should deeply look into these issues, supporting the creation of moral rules and regulation systems that find a middle ground between benefits from analyzing information and protection for both personal rights plus social values.

In the time to come, these are the areas that researchers can explore more. It will help in making a complete understanding of digital environment and its effect on social, economic, and technical progress. To deal with complex difficulties that emerge in rapidly changing digital area, it is important to involve in cooperative efforts which include teams having varied fields of knowledge.

References

[1] World Wide Web Size. Available online: https://www.worldwidewebsize.com/.

[2] Internet Statistics: Number of Users & Websites in 2022. Forbes. Available online: https://www.forbes.com/home-improvement/internet/internet-statistics/#:~:text=There%20are%205.35%20billion%20internet,the%20internet%2C%20according%20to%20Statista.

[3] "Digital 2022: Azerbaijan." DataReportal. Available online: https://datareportal.com/reports/digital-2022-azerbaijan

[4] "Graph Analytics." NVIDIA. Available online: https://www.nvidia.com/en-us/glossary/graph-analytics/

[5] Khder, Moaiad. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing and its Applications. 13. 145-168. 10.15849/IJASCA.211128.11. https://www.researchgate.net/publication/357401723_Web_Scraping_or_Web_Crawling_State_of_Art_Techniques_Approaches_and_Application

[6] Zhao, Bo. (2018). "Web Scraping." Available online: https://www.researchgate.net/profile/Bo-Zhao-3/publication/317177787_Web_Scraping/links/5c293f85a6fdccfc7073192f/Web-Scraping.pdf

[7] Chen, Y., Guan, Z., Zhang, R. *et al.* A survey on visualization approaches for exploring association relationships in graph data. *J Vis* **22**, 625–639 (2019). https://doi.org/10.1007/s12650-019-00551-y

[8] W. Xing and A. Ghorbani, "Weighted PageRank algorithm," Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004., Fredericton, NB, Canada, 2004, pp. 305-314, doi: 10.1109/DNSR.2004.1344743. keywords: {Web mining;Web pages;Computer science;Niobium;Content based retrieval;Uniform resource locators;Electronic learning;Pattern analysis;Topology;Communication networks},

[9] Stothers JAM, Nguyen A. Can Neo4j Replace PostgreSQL in Healthcare? AMIA Jt Summits Transl Sci Proc. 2020 May 30;2020:646-653. PMID: 32477687; PMCID: PMC7233060.

[10] "Betweenness Centrality." Neo4j. Available online: https://neo4j.com/docs/graph-data-science/current/algorithms/betweenness-centrality/
[11] "Eigenvector Centrality." Neo4j. Available online: https://neo4j.com/docs/graph-data-science/current/algorithms/eigenvector-centrality/
[12] "Degree Centrality." Neo4j. Available online: https://neo4j.com/docs/graph-data-science/current/algorithms/degree-centrality/
[13] "PageRank." Neo4j. Available online: https://neo4j.com/docs/graph-data-science/current/algorithms/page-rank/
[14] Krotov, Vlad & Johnson, Leigh. (2022). Big web data: Challenges related to data, technology, legality, and ethics. Business Horizons. 66. 10.1016/j.bushor.2022.10.001.
[15] "Web Scraping Challenges in E-commerce." DataHen. Available online: https://www.datahen.com/blog/web-scraping-challenges-in-ecommerce/
[16] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. 1998. Graph Drawing: Algorithms for the Visualization of Graphs (1st. ed.). Prentice Hall PTR, USA.
[17] Raga'ad M. Tarawaneh, Patric Keller, and Achim Ebert. A General Introduction To Graph Visualization Techniques. In Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011. Open Access Series in Informatics (OASIcs), Volume 27, pp. 151-164, Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2012) https://doi.org/10.4230/OASIcs.VLUDS.2011.151
[18] Eades P. A heuristic for graph drawing. Congressus numerantium. 1984 May;42(11):149-60.
[19] Malaga, R. A. (2008). Worst practices in search engine optimization. Communications of the ACM, 51(12), 147-150.
[20] Jakub Zilincan. CBUINTERNATIONAL CONFERENCE ON INNOVATION,TECHNOLOGY TRANSFER AND EDUCATIONMARCH25-27,2015,PRAGUE,CZECH REPUBLIC https://doi.org/10.12955/cbup.v3.645
[21] Harto, A. B. (2019). Implementing Website Design Based on Search Engine Optimization (SEO) Checklist to Increase Web Popularity. *Journal of Applied Information, Communication and Technology*, *6*(2), 87-97. https://doi.org/10.33555/ejaict.v6i2.67
[22] Hao, M.C., Dayal, U., Hsu, M., Sprenger, T., Gross, M.H. (2001). Visualization of directed associations in e-commerce transaction data. In: Ebert, D.S., Favre, J.M., Peikert, R.

(eds) Data Visualization 2001. Eurographics. Springer, Vienna. https://doi.org/10.1007/978-3-7091-6215-6_20

[23] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools https://graphviz.org/documentation/EGKNW03.pdf