

**Ministry of Higher Education and Scientific Research
University of Mosul
College of Computer Science and Mathematics
Department of Computer Science**



A Cloud Big Data Parallel Computing Approach for Academic Capital Dynamics of the Scientific Research in Iraq

**A Thesis Submitted to the Council of the College of
Computer Science and Mathematics
University of Mosul
as a Partial Fulfillment of Requirements
for the Degree of Doctor of Philosophy in
Computer Science**

**By
Nagham Ajeel Sultan Al-Ajeely**

**Supervised by
Prof. Dr.Dhuha Basheer Abdullah Albazaz**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿رَبِّ أَوْزَعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَى
وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ وَأَدْخِلْنِي بِرَحْمَتِكَ فِي عِبَادِكَ
الصَّالِحِينَ﴾

صَدَقَ اللَّهُ الْعَظِيمُ

سورة النمل
الآية (١٩)

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Computer Science, College of Computer Science and Mathematics, University of Mosul, as a partial fulfillment of the requirements for the Degree of **Doctor of Philosophy in Computer Science**.

Signature:

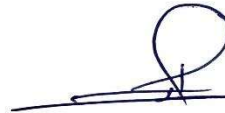
Name: Prof. Dr Dhuha Basheer Abdullah

Date: / / 2024

Report of the Linguistic Evaluator

I certify that the linguistic evaluation of this thesis was carried out by me and it is accepted linguistically and in expression.

Signature:



Name: Dr. Aws Abbas Younis

Date: / / 2024

Report of Director of Postgraduate Studies Committee

According to the recommendations presented by the supervisor and linguistic evaluator of this thesis, I nominate this thesis to forward for discussion.

Signature:

Name: Prof. Dr. Baydaa Ibrahim Khalil

Date: / / 2024

Report of the Head of the Department

According to the recommendations presented by the supervisor of this thesis, linguistic evaluator, and the Director of Postgraduate Studies Committee, I nominate this thesis to forward for discussion.

Signature:

Name: Dr. Wael Wadullah Mahmood

Date: / / 2024

Committee Certification

We, certify that we have read the entitled thesis (**A Cloud Big Data Parallel Computing Approach for Academic Capital Dynamics of the Scientific Research in Iraq**) and as Examining Committee, we have examined the student in its contents on **25 / 4 / 2024** and that, in our opinion, it is adequate as a thesis for the Degree of **Doctor of Philosophy in Computer Science**.

Signature:

Name: Dr. Murtadha M. Hamad
Chairman

Date: / / 2024

Signature:

Name: Dr. Najla Badie Ibraheem
Member

Date: / / 2024

Signature

Name: Dr. Saja J. Mohammed
Member

Date: / / 2024

Signature

Name: Dr. Mohammed Zaki Hasan
Member

Date: / / 2024

Signature

Name: Dr. Ashraf A. Abdulmajeed
Member

Date: / / 2024

Signature

Name: Dr. Dhuha Basheer Abdullah
Member & Supervisor

Date: / / 2024

Acknowledgement

To begin with, I must thank Allah the Most Gracious, Most Merciful for completing this thesis.

Foremost, I would like to express my sincere gratitude to my supervisor Prof. *Dr Dhuha Basheer Abdullah*, Dean of the college of computer science and mathematics, for her continuous support in my PhD study and research and for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the research and writing of this thesis. I had so many ways to get to my destination, but she showed me the right one. I am really thankful to her, so she is truly a wonderful and sincere person in her hard work, for a person's debt is not a virtue, but repayment in it is in itself a virtue.

Also, My thanks are extended to the staff of the computer science department who were helpful to me when needed.

Finally, my deepest gratitude goes to all my family members, who have never stopped encouraging and supporting me. They have always supported me with their valuable feedback.

ABSTRACT

The massive usage of modern technologies and the Internet with smart devices yields large-scale data called Big Data. The development of cloud computing technologies enables handling big data in terms of processing and storage; therefore, several techniques have been developed by companies and institutions aiming to come up with efficient approaches. Previous works optimizing these approaches can be performed by involving advanced technologies such as parallel processing, containerization, and virtualization.

Moreover, in sociology, the concept of social capital represents an individual's collective behavior, and norms in a community. In the same context, academic capital means the collective performance of an author or an institution. This concept, has not been studied in previous literature. On the other hand, Google Scholar is considered the most well-known web search engine for accessing papers around the world. Its data can be used to assess the academic performance of a particular author, university, or institution with citations, h-index, publishers, etc. However, processing, analyzing, and visualizing these data need to adopt dedicated dynamic methods. Therefore, developing an dynamic efficient approach to convert big data to knowledge for evaluation by authors or universities is critical.

This thesis introduces a dynamic system by proposing a novel rank called Academic Capital Distributed Network (ACDN). It dynamically collects extensive data from Google Scholar using several Cloud Computing technologies. The collected data was then processed, visualized, and analyzed to assess the dynamic academic performance of Iraqi authors and universities.

Web scrapers were designed to gather dynamic data from Google Scholar of Iraqi universities and authors based on their official universities domains. These scrapers are based on concepts inspired by parallel processing techniques using dedicated virtual machines. The collected data is seamlessly stored using a Replica Cluster in MongoDB Atlas to facilitate efficient data management and provide a robust and scalable dynamic infrastructure for storing and managing the collected data. This method offers high availability, fault tolerance, read scalability, and data integrity. Moreover, a Cloud-based parallel computing environment is adopted to facilitate the efficient organization and processing of the collected data, leveraging big data frameworks, with a particular focus on Apache Spark.

Two novel datasets have been created: one for authors and one for universities. By using these datasets, two network models were generated; the first is for Iraqi universities; termed as Iraqi Universities Network (IUN), and the latter is for Iraqi authors; termed as Iraqi Scholars Network (ISN). Academic capital of authors and universities was calculated using data from the two network models IUN and ISN, and Google Scholar indicators. Furthermore, the performance of the Iraqi authors and universities was evaluated also, ranks were assigned based on their respective metrics of academic capital which change dynamically in response to the indicators and search criteria employed by the author or university.

The findings show the efficiency of the proposed approach in terms of data quality and reliability in dynamic collecting big data from Google Scholar Cloud. The results also offer interesting facts about the Iraqi universities and authors. In addition, collaboration patterns among authors and universities were also extracted using the two network models. The proposed rank was also benchmarked with other international ranks. The main results show that the average degree of the ISN is 1.8, which means that each Iraqi scholar has approximately coauthored two papers on average. This result is considered acceptable compared to ACM international network. On the other hand, the average shortest path length of IUN is 3.2 meaning that the scientific collaboration among the Iraqi universities indicates a moderate distance between nodes on average. However, the clustering coefficient of IUN is 0.249, which reflects that the Iraqi universities are not strongly connected and the tendency to cluster together is low (far from 1). Furthermore, The results of Academic Capital of authors show that a Computer Science author from University of Babylon has the highest Academic Capital among all the authors in the Iraqi universities. The Academic Capital of the Iraqi universities show that the University of Baghdad has the highest value followed by Al-Mustansiriyah University, University of Basrah, University of Mosul, University of Babylon, and so on. This result is relatively in agreement of the other international ranks such as QS, Times, Webometrics. This work is considered the first kind of analysis that rates all the Iraqi universities and authors as the main focus, which is of interest to the Ministry of Higher Education and Scientific Research officials in Iraq.

Table of Contents

CHAPTER ONE: Introduction.....	1
1.1 Background	1
1.2 Related Works	3
1.3 Problem Statement	10
1.4 Objectives.....	11
1.5 Contributions	12
1.6 Thesis Layout	13
CHAPTER TWO: General Background	14
2.1 Preface	14
2.2 Academic Capital	14
2.2.1 Google Scholar	15
2.2.2 Bibliographic properties of GS as a database	17
2.3 Information Retrieval	18
2.3.1 Web mining technology.....	19
2.3.2 Web Scraping	21
2.3.3 Web Scraping Architecture.....	22
2.3.4 Web Scraping Techniques	23
2.3.5 Web Scraping tools.....	24
2.4 Computing Paradigms and Big Data	26
2.4.1 Big Data Architecture Components.....	27
2.4.2 Big Data Characteristics	28
2.4.3 Big Data Processing Frameworks.....	29
2.4.4 Cluster Computing.....	31
2.4.4.1 Cluster Computing Architecture	31
2.4.4.2 Cluster Computing Types	32
2.4.4.3 Cluster Computing and Big-Data Analytic Platforms	33
2.5 Cloud Computing	34
2.5.1 Cloud Computing Layers.....	35

2.5.2 Cloud Computing Deployment Models.....	37
2.5.3 Cloud Characteristics.....	38
2.6 Parallel Processing	39
2.7 Virtualization Concepts.....	40
2.8 Complex Network and Graph Theory	42
2.9 Complex Network Visualization Platforms	43
2.9.1 Cytoscape.....	43
2.9.2 NetMiner.....	43
2.9.3 Gephi	44
2.9.4 Pajek	44
CHAPTER THREE: System Frameworks and Tools	45
3.1 Preface	45
3.2 SerpApi.....	45
3.2.1 Design of SerpApi Strategy	46
3.3 Big data Framework and Parallel Computing	47
3.3.1 Parallel Data Processing System	47
3.3.2 Apache Spark.....	50
3.3.2.1 Spark architecture	50
3.3.2.2 Apache Spark Components.....	52
3.3.2.3 Apache Spark Cluster Management.....	54
3.3.3 MongoDB Database	55
3.3.3.1 Architecture of MongoDB Deployment	56
Figure 3.4 Architecture of Sharded MongoDB (Shin et al., 2022).....	57
3.3.3.2 Clustering in MongoDB Atlas	58
3.3.4 Integrating Spark with MongoDB	59
3.4 Cloud Service and Distributed Systems	61
3.4.1 AWS Cloud Service.....	63
3.4.1.1 Platform as a Service (PaaS).....	63
3.4.1.2 Infrastructure as a service (IaaS).....	64
3.4.1.3 Software as a Service (SaaS)	64

3.5 Virtualization Concepts	64
3.5.1 Virtualization Tools (VirtualBox)	65
3.5.2 Virtualization in Cloud Computing (EC2).....	66
3.5.2.1 Amazon EC2 Main Feature	68
3.6 Complex Network	68
3.6.1 Complex Networks Measurements (Network Level Metric).....	69
3.6.2 Complex Networks Measurements (Nodes Level Metric)	70
3.6.3 Collaboration Networks	72
3.6.4 Co-authorship network.....	72
3.7 Visualization.....	73
3.7.1 Cytoscape Software	73
CHAPTER FOUR: ACDN System Design Methodology	75
4.1 Preface	75
4.2 System Architecture	75
4.3 System Strategy	77
4.4 ACDN Proposed System	79
4.4.1 Data Collection	80
4.4.1.1 Unischolar Model.....	80
4.4.1.2 ScholarSync IRAQ Model	84
4.4.2 Processing Data with Cluster Computing and Storage.....	89
4.4.2.1 Data Storge.....	89
4.4.2.2 Data organization and Database Deployments	90
4.4.2.3 Data Transmission after Preliminary Processing.....	92
4.4.2.4 Data Parallel Processing with Apache Spark.....	93
4.4.2.5 Deployment of Spark on AWS EC2 Cloud Cluster.....	96
4.4.3 Adopting Parallelism in Proposed System	99
4.4.4 Adopting Virtualization in Proposed System	100
4.4.5 Data set formation	102
4.4.5.1 Iraqi Scholar Network Model (ISN)	103
4.4.5.2 Iraqi University Network Model (IUN)	103

4.4.6 Network Visualization	104
4.5 Academic Capital Distributed Network (ACDN)	104
CHAPTER FIVE: Implementation and Results Discussion	110
5.1 Preface	110
5.2 Hardware and Software Requirements	110
5.3 System implementation	112
5.3.1 Data Collection Steps	113
5.3.1.1 Unischolar Model.....	114
5.3.1.2 ScholarSync Model.....	118
5.3.1.3 Overcoming Challenges in Scholarly Data Scraping.....	121
5.3.2 Managing and Storing Raw Data.....	124
5.3.3 Parallel Data Processing	128
5.3.4 Procedure for Data Processing Output	132
5.3.5 Performance Evaluation.....	136
5.3.5.1 ISN Performance Evaluation	136
5.3.5.2 IUN Performance Evaluation.....	151
5.3.6 Academic Capital	163
CHAPTER SIX: Conclusions, and Future Works	170
6.1 Preface	170
6.2 Conclusions	170
6.3 Future Works	171
Refrence.....	173

List of Figures

Number	Title	Page
2.1	GS, a Big Data Bibliometric Tool	18
2.2	Methods of Information Retrieval	19
2.3	The Web Scraping Basic Architecture	22
2.4	Web Scraping Phases	23
2.5	Numerous Big Data Architecture Elements	28
2.6	Characteristic of Big Data	29
2.7	Big Data Framework	30
2.8	Cluster Computing Architecture	32
2.9	Simple Architecture of Cloud Computing	35
2.10	Cloud Computing Services	37
2.11	Hypervisor-Based Virtualization Architectures	42
3.1	Apache Spark Architecture	52
3.2	Components Apache Spark Ecosystem	54
3.3	Replication of Data in Memory and Disk in MongoDB	57
3.4	Architecture of MongoDB	58
3.5	Deploy a MongoDB Cluster	59
3.6	Architecture of MongoDB Spark Connector (SDBC)	61
3.7	Architectural of AWS	63
3.8	Architectural of VirtualBox	66
3.9	EC2 Architecture	68
4.1	The Proposed Architecture System	77
4.2	System Work Strategy	79
4.3	Steps of Unischolar Model	84
4.4	Steps of ScholarSync IRAQ Model	89
4.5	Deploying a Database in MongoDB Atlas	92
4.6	Integration Between MongoDB and Apache Spark	93
4.7	Spark Cluster Architecture	94
4.8	Spark Core Configuration	95
4.9	Parameters to Configure Apache Spark Ecosystem	97
4.10	Configure Spark Parameters	97
4.11	Architecture Based in Apache Spark Layer	99
5.1	System Implementation Architecture	112

5.2	Sample of Identifiers for the Iraqi Universities	115
5.3	Sample Data from a Specific University	116
5.4	5.4a Shows All Authors in as Specific University 5.4b Shows Information and Articles of Each Author in Specific Uuniversity	117
5.5	Sample Data of Google-Scholar-Profiles (edu.iq)	119
5.6	Researcher's Information by Iraqi Domain	121
5.7	Support from SerpApi Team with 60,000 searches	123
5.8	Store Streaming Data in Cloud	125
5.9	Configuration of Replica Cluster	125
5.10	Network Indicators in Cluster	126
5.11	Logical Size Indicators in Cluster	126
5.12	Connections Indicators in Cluster	126
5.13	Parameters for the Primary and Secondary Node	128
5.14	Running Cluster	129
5.15	Utilize Spark in the Cloud	130
5.16	Launch the Workers and Master nodes	131
5.17	5.17a, Sample of DataSet After Processing 5.17b, Sample of DataSet After Processing	134- 135
5.18	The Frequency of Papers Published by Iraqi Authors (Grouped by Universities)	136
5.19	Pareto Rule Interpretation Regarding the Iraqi Authors	137
5.20	The Frequency of Citations of the Iraqi Authors (Grouped by Universities)	138
5.21	The Frequency of the h-index of the Iraqi authors (Group By Universities)	138
5.22	The Frequency of the i10-index of the Iraqi authors (Group by Universities)	139
5.23	Visualization of ISN Network of Local and International Scholars	140
5.24	Distribution of Betweenness Centrality	142
5.25	Distribution of Degree Centrality	143
5.26	Distribution of Closeness Centrality	143
5.27	Visualization of ISN of Iraqi Authors Only	145
5.28	Betweenness Centrality Distribution	146
5.29	Degree Distribution Centrality	147
5.30	Eigen Centrality Distribution	147
5.31	Closeness Centrality Distribution	148

5.32	ISN Performance in term of h-index	149
5.33	ISN Performance in term of i10-index	150
5.34	Visualizing Universities as Bars	152
5.35	A close view of the collaboration among the Iraqi universities	152
5.36	A Closer View of the Collaboration Among the Iraqi Universities	153
5.37	A Microscopic View to the collaboration Among the Iraqi Universities	153
5.38	Geo Layout Visualization of The Iraqi Universities alongside their Collaborations with Other Iraqi Universities	156
5.39	Degree Distribution	157
5.40	Betweenness Centrality Distribution	157
5.41	Closeness Centrality Distribution	158
5.42	Clustering Coefficient Distribution	158
5.43	IUN Visualization After Removing the Weakest 25% Edges	159
5.44	IUN Visualization After Removing the Weakest 50% Edges	160
5.45	IUN Visualization After Removing the Weakest	160
5.46	IUN Visualization of The Top 20% Collaboration	161
5.47	Distribution of Modularity in IUN Communities	162
5.48	Authors' Academic Capital Distributions	164
5.49	Academic Capital of the Iraqi Universities	166

List of Tables

Number	Title	Page
2.1	A Comparison Between Web Scraping Vs. Web Crawling	21
2.2	Comparison of Cloud Computing Model	38
5.1	Hardware Requirements	111
5.2	Authors Network	141
5.3	Comparison between the Authors' Network and Other Similar Networks	142
5.4	ISN Network Characteristics After Removing the International Scholars	144
5.5	Characteristics of the IUN Network	154
5.6	Community Detection Using Louvain and Leiden	163
5.7	Top 10 Iraqi Authors in term of Academic Capital	165
5.8	Benchmarking the Proposed Academic Capital-Based Rank With Other Local and International Ranks	167

List of Abbreviations

Abbreviation	Definition
AC	Academic Capital
ACC	Average Clustering Coefficient
ACDN	Academic Capital Distributed Network
ACDN	Academic Capital Distributed Network
ACM	Association for Computing Machinery
Amazon EBS	Amazon Elastic Block Store
AMIs	Amazon Machine Images
API	Application Programming Interface
AWS	Amazon Web Services
BD	Big Data
C	Citations
CB	Betweenness Centrality
C_c	Closeness Centrality
C_o	Clustering Coefficient
CPM	Constant Potts Model
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
D	Degree Centrality
DC	Degree Centrality
DAG	Directed Acyclic Graph
DBaaS	Database as a Service
DFS	Distributed File System
DOM	Document Object Model
DW	Data Warehouse
ETL	Extraction, Transformation, Loading
E	Eigen Centrality
EC2	Elastic Compute Cloud
GS	Google Scholar
GSC	Google Scholar Citations
GSM	Google Scholar Metric
GUI	Graphical User Interface
H10	H10-Index

HDFS	Hadoop Distributed File System
HPC	High Performance Computing
HI	H-Index
IaaS	Infrastructure as a service
ICT	Information and Communications Technology
IE	Information Extraction
IR	Information Retrieval
ISN	Iraqi Scholar Network Model
IT	information technology
IUN	Iraqi University Network Model
JSON	Javascript Object Notation
JSPS	Japan Society for the Promotion of Science
JVM	Java Virtual Machine
L	Average Path Length
MLlib	Machine Learning Library
MongoDB	Mongo Data Base
MPI	Message Passing Interface
NASA	Network Analysis Software Applications
NFAT	Network Forensic Analysis Tool
NLP	Natural Language Processing
OS	Operating System
PaaS	Platform as a Service
PBRA	Passphrase Based Rest Api
PVM	Parallel Virtual Machine
QEMU	Quick Emulator
RDD	Resilient Distributed Dataset
SaaS	Software as a Service
SDBC	MongoDBConnector for Spark
SNA	Software Network Analysis
SP	Scientific Publications
SQL	Structured Query Language
VMM	Virtual Machine Manager
VMs	Virtual machines
VOS	Visualization of Similarity
YARN	Yet Another Resource Negotiator

CHAPTER ONE

Introduction

1.1 Background

In today's rapidly evolving and interconnected world, academic capital has emerged as a crucial determinant of an individual's potential to succeed and advance in society. Coined by Pierre Bourdieu in 1979, the concept of academic capital encompasses collective knowledge, skills, and experiences acquired through education and academic pursuits. It represents a valuable resource that individuals and institutions can leverage to navigate the complex landscape of academia, research, and professional growth. Academic capital can be defined as the potential possessed by individuals, institutions, universities, or colleges to advance and succeed in academia and society at large. It represents the cumulative value derived from educational achievements, experiences, knowledge, skills, and intellectual contributions within the academic field. Just as financial capital represents wealth, and social capital represents social networks and connections, academic capital encompasses the assets and resources acquired through academic pursuits (Royal, 2022).

Google Scholar plays a significant role in shaping academic capital, particularly in scientific research. It enhances visibility and access, allowing researchers and institutions to showcase their work and gain recognition. Google Scholar's citation metrics provide indicators of research impact, contributing to academic capital by demonstrating the influence of one's work. The platform also facilitates collaboration and networking, helping researchers connect with peers and foster interdisciplinary approaches. Additionally, Google Scholar is used for research assessment, with strong citation records and high impact enhancing academic capital in evaluations (Delgado López-Cózar et al., 2019).

The number of Internet users and the volume of data they generate, mostly unstructured data, has increased dramatically as a result of the adoption of new technologies. Web scraping is a technique that big data analysts should consider, as it automatically gathers information from webpages. Thousands of variables can be found in a web scraping dataset in just a few hours, web scraping can also produce tens of thousands of instances in a smaller and more

manageable dataset (Khder, 2021). When conducting data extraction on a large scale, combining web scraping with other technologies like Bigdata might produce additional value. Furthermore, it is an excellent method for obtaining unstructured data from websites and converting it into structured data that can be kept and examined in a database. Other names for web scraping include screen scraping, online data harvesting, web data extraction. Data mining includes web scraping, nevertheless, extracting information from a website and converting it into a comprehensible format, such as a database, spreadsheet, Comma-Separated Values (CSV) file, or JSON file, is the main objective of the web scraping process (Vanden Broucke & Baesens, 2018).

In recent years, the Hadoop ecosystem has embraced Apache Spark as a potent framework for quicker distributed computation on large data sets. It achieves this by utilizing in-memory primitives, which in some cases enable it to operate 100 times quicker than Hadoop. The most fundamental aspects of big data technology are its processing, accessing, securing, and storage capabilities. At the core of this Bigdata is Spark, an open-source cluster (in-memory) computing platform and unified data processing engine that is faster and more dependable than previous generations when it comes to cutting-edge analysis of all kinds of data. It is powerful in joining various datasets from many disjointed data sources (Inoubli et al., 2018).

Spark offers MLlib, GraphX, Spark Streaming, and Spark SQL for processing structured, semi-structured, and unstructured data. It is changing quickly, due to updates to its fundamental APIs, and the addition of higher-level libraries. Better performance in the design of scalable data pipelines and algorithms is made possible by its central data abstraction, that is, the Resilient Distributed Dataset (RDD). A range of operators and the RDD's effective data sharing allow for the effective design and implementation of various workloads. Spark applications can be executed in Java, Python, and Scala and it provides solutions for data processing that have high scalability and improved performance (Salloum et al., 2016).

Utilizing Network Analysis Software Applications (NASA), which analyze, visualize, identify, or recognize nodes (learning, knowledge, organizations) and edges (association or relationship or connections) from various data formats and sources, can accomplish Software Network Analysis.

Many NASA tools are available for manipulating and visualizing "Big Data," as well as for displaying social media network analyses of the majority of academics and other users to facilitate analysis. These different NASA are used to share or retrieve the dataset, which may then be analyzed further to comprehend the final dataset (Mittal et al., 2019).

In order to generate edges and node lists, adjacency matrices, graph visualizations, and other things, different NASA are used to compute network statistics and attributes (such as radius, diameter, Betweenness, shortest path, community detection, etc.). Understanding network dynamics is made simple by graph representation or visualization, which also makes it easier to find relationships, timelines, isolated nodes, strong and weak communities, etc. Measures of network graphs include betweenness measurement, diameter, density, centrality, and the degree of hubs, among others. Cytoscape and Gephi's adaptable graphical user interface is used to view, manage, and analyze networks. It is a strong tool with a vibrant developer community that generates graphs more effectively than NASA-NodeXL. This program includes a three-dimensional rendering engine that allows it to display big and intricate real-time graphs quickly, in addition to the ability to calculate network metrics like average diameter and clustering coefficients (Mittal et al., 2019).

1.2 Related Works

This section contains an overview of the previous studies in several areas related to the thesis.

The technique and possible applications of co-authorship network analysis in health research are covered by Paula (Fonseca et al., 2016). The work procedure carries out co-authorship network analysis in health research and emphasizes how it can be used to evaluate collaboration patterns and pinpoint top researchers and institutions. It outlines standard network metrics that may be used to study co-authorship networks, and gives an overview of the global research network for vaccines against the Chikungunya virus as an example of how the method is applied. This work employs the following methods: retrieving scientific publications, standardizing author and organization entries; network visualization and metric computation, and outcome interpretation. In health research, co-authorship network analysis is carried out using these techniques. This work provides common network metrics that can be used to study co-authorship networks and outlines the fundamental procedures for

carrying out co-authorship studies in health research. It uses publication records gathered from organized bibliographic databases as its data source.

The authors (Shi & Zhang, 2017) proposed a scalable distributed web crawler system based on Hadoop technology, aiming to address the challenges posed by the large data era. The system design leverages the distributed and Cloud Computing features of Hadoop to achieve high scalability. The authors compared their system with existing mature systems and highlighted its superiority in terms of data statistics, particularly in the context of massive data. The system involves a simple pretreatment of the crawled pages, converting HTML code to XML using the HtmlToXmlDriver module. The crawling process follows a basic flow, starting with selectin of the URL seed file and uploading it to the Hadoop cluster's distributed file system (HDFS). The nodes in the system handle URL resolution and store the resolved URLs in a database for further processing by the master node.

The authors Kim et al. (Khalil & Fakir, 2017) describe the design and functionality of RCrawler, which is a contributed R package for domain-based web crawling and content scraping. The authors implemented RCrawler in an R environment and optimized it to handle the limitations of R. The main methods used in RCrawler are the development of a parallel web crawler in the R environment and the implementation of multi-threaded crawling, content extraction, and duplicate content detection. They included of functionalities, such as URL and content-type filtering, depth level controlling, and a robot.txt parser. The results demonstrate the effectiveness and flexibility of RCrawler as a tool for web crawling and content scraping, particularly for those working in the R environment. The limitation does not take ethical issues in considerations of web crawling and content scraping, such as, issues related to privacy and data ownership.

Gupta (Gupta, 2018) conducts a performance evaluation of web collection data in NoSQL databases, specifically Cassandra and MongoDB, focusing on scalability and security aspects. This work presents a performance evaluation of web collection data in data stores, such as NoSQL-Cassandra and MongoDB, for scalability of applications and proposes a Passphrase Based REST API (PBRA) model to deal with highly unstructured big data emerging from the Twitter social networking service. The authors use AWS CMT to notify

changes in data and OpsCenter to monitor the performance of the data stores. The results show that though the time taken to load and retrieve bulk data records is higher than dealing with ciphertext, MongoDB performance better than Cassandra with the proposed security model. The work does not provide a real-time implementation of the security modules at data files level or client interface levels.

Fujita (Fujita, 2019) proposes a quantitative method to evaluate researchers in organizational R&D fields based on network centralities of co-author networks by Masanori Fujita. The paper emphasizes the importance of finding and encouraging young promising researchers for the development of science and technology. The proposed method focuses on evaluating researchers' competencies in both the expertise of the research fields and cooperativeness with others in projects. The method involves analyzing the network centralities of co-author networks, which is measured from the academic literature database. The research provides insights into evaluating researchers in organizational R&D fields using network analysis techniques, which can be valuable for researchers and institutions involved in such projects. The paper concludes that the proposed method can stand as a practical approach to evaluate researchers in organizational R&D fields. The proposed method was tested on a dataset of academic papers in only two fields, life science and information technology

This paper done by Shaikh (Shaikh et al., 2019) provides a technical review on big data analytics using Apache Spark, highlighting its use in emerging technologies and its architecture, hardware requirements, ecosystem, and features presented by Eman Shaikh. This paper highlights that Apache Spark's in-memory computation and data parallelism make it remarkably faster than other corresponding frameworks, which has practical implications for organizations dealing with large-scale data processing. Apache Spark has the potential to enhance big data analytics and enable faster, more efficient, and real-time data processing for various applications and industries.

This work of dynamic data auditing using MongoDB in a cloud platform by Akilandeswari et al. (Akilandeswari et al., 2019) focuses on addressing concerns related to data safety and integrity in shared computing resources. The authors highlight that existing remote data auditing techniques often incur

computational overhead and are primarily designed for static datasets. To overcome these limitations, the proposed system utilizes dynamic data updating and MongoDB as the database. The research emphasizes the use of JSON format for stored files in the database, which enables efficient identification of updated data and reduces computation time. Additionally, RSA signatures are employed to ensure data confidentiality. This paper suggests that the proposed technique can enhance speed, efficiency, and overall trust in the cloud space by reducing the cost and computational power requirements of third-party auditors.

Retrieving data from World Wide Web requires data mining techniques. The authors Javed et al (Shamrat et al., 2020) discussed the development and implementation of a web crawling system using Python programming language and its default libraries to retrieve data from the World Wide Web (WWW). The system is capable of extracting information from web pages and filtering it for usable and graphical purposes for users. The system explains the implementation of the web crawler and its functions, but does not consider the impact of web crawling on the web servers and the potential risks of overloading them.

The authors Iskandar et al. (Iskandar et al., 2020) contributed to the field of bibliometric analysis by utilizing mathematical data to visualize patterns in the field of science, specifically in the context of COVID-19 research. The paper discusses co-authorship of COVID-19 research, covering author productivity and author collaboration. The data is collected from the Web of Science Core Collection database using specific search criteria, including the keyword "covid" and the year of publication 2020, and analyzed using VosViewer software. The analysis was conducted at the author and country level, examining the co-authorship relationships among researchers in the field of COVID-19 research. The authors aimed to identify influential writers and patterns in the field of COVID-19 research through their bibliometric analysis. The VosViewer program is used to visualize the bibliometric analysis. Co-authorship is a type of analysis where authors and countries are the unit of analysis. The analysis is limited to the unit of analysis of authors and countries, and other units of analysis such as institutions, journals, and keywords are not included in this paper. The collected COVID-19 research literature was collected from the WoS database.

The work presented by Aung et al. (Aung & Nyunt, 2020) focused on the analysis of community structure in scientific collaboration networks, specifically in the co-authorship network of Computer University in Myanmar. The Louvain algorithm is utilized in this paper to examine the structure of the community in the co-authorship network and highlights that the Louvain algorithm is simple, easy to implement, and efficient in recognizing community structures in large networks. Neo4j, which is a graph database management system, is used to visualize the results of the co-authorship network analysis. The paper also mentions the use of modularity as a measure of the quality of the cluster structure identified by community discovery algorithms. In general, the research demonstrates the effectiveness of the Louvain algorithm in identifying qualitative community structures in co-authorship networks and showcases the visualization capabilities of Neo4j in analyzing scientific collaboration networks.

Higaki et al. carries out a co-authorship network study on 2857 research articles in the field of cardiovascular medicine between 2009 and 2019 (Higaki et al., 2020). The Web of Science provides the bibliographic information for the analysis, and the co-authorship networks are displayed as undirected multigraphs. In the context of applying machine learning to cardiovascular disease research, a number of network measures, including number of communities, average degree, clustering coefficient, and network density, are computed to evaluate the composition of collaborative networks. Centrality metrics are used to identify the leading authors. Additionally, a Monte Carlo simulation employing the Barabasi-Albert model is used to examine the significance of the co-authorship network's features in the largest component. Planning upcoming scientific collaborations in the field of cardiovascular disease research can benefit from the analysis's conclusions. Since the paper only looks at data from 2009 to 2019, its conclusions cannot accurately represent the state of machine intelligence-based collaborative networks in cardiovascular care today.

The author Junior et al. (Pessoa Junior et al., 2020) focused on interdisciplinary collaborations in the Brazilian scientific community. The authors collected data from the Lattes Platform, which is an initiative from CNPq, and they analyzed the coauthorship network to characterize these collaborations. They found that 35.2% of all collaborations are

interdisciplinary, and 57.6% of researchers had participated in at least one interdisciplinary collaboration. The paper also investigated the intensity of interdisciplinary collaborations across different communities and explored collaboration patterns over time. This research provides valuable insights into the growth and interdisciplinary nature of the Brazilian coauthorship network, highlighting the importance of bridging academic relationships in strengthening scientific communities. The paper relies on data from the Lattes Platform, which may not capture all scientific collaborations or accurately reflect the interdisciplinary nature of some collaborations.

Fujita et al. (Fujita et al., 2021) Research on evaluating young researchers using network centralities of co-authorship networks was presented from the academic literature. They suggest a novel use of the betweenness centrality measure with co-author networks for the assessment of young scientists who show promise, particularly in biological sciences. By contrasting it with more established indices, such as, the h-index, the authors confirm the usefulness of this metric as an index for assessing junior researchers. Through their analysis of 1.92 million articles in the field of biological science, researcher's fellows of the Japan Society for the Promotion of Science (JSPS) ranked higher and advanced faster than other researchers. By comparing the suggested method to the h-index and centralities from current literature, they also highlighted differences at early stages between JSPS Research Fellows and other researchers. Similar evaluations in other study domains such as bioengineering, bioinformatics, or astrophysics are proposed by the authors, who think that the betweenness centrality measure could be an excellent tool for identifying bright young researchers.

The work by Afolabi, et al. (Afolabi et al., 2021) studies properties of co-authorship network in the Computer Science department of Covenant University in Nigeria and aims to bridge the gap in scientific recommendations for improving the quality and quantity of computing performance in Nigerian universities. It explores network properties such as centrality measures and network coefficients to understand the collaboration patterns among researchers. Based on the analysis, the authors develop an academic collaboration recommendation system using the Adamic-Adar Index measure for link prediction. This system is designed to enhance mentorship and facilitate idea exchange within the academic community, ultimately improving the

performance of computer science academics in Nigeria. The research findings provide valuable information on the centrality values of researchers in computer science, forming the foundation for the development of the collaboration recommendation system. This paper emphasizes the potential of this system to create a small world research network and enhance the quantity and quality of academic performance in computer science in Nigeria.

Chi et al. (Chi et al., 2021) proposed the design and implementation of a hotel big data analysis platform using Hadoop and Spark technology. The authors collect basic data and user comment data of city hotels in Qingdao, Shandong Province, using a crawler program. They upload the data to the Hadoop storage platform and preprocess it using Spark for big data analysis. They also use Hive to build a big data platform data warehouse and Echarts for visual display and analysis of the results. The authors explain that this is an important area of research due to the increasing amount of data generated by the hotel industry, including customer reviews, booking data, and operational data. The platform is designed to enable effective data processing, analysis, and visualization of this data, with the aim of providing insights for decision-making and improving hotel operations. The platform includes several components, such as data processing using Hadoop, data analysis using Spark, and visualization using Tableau. The authors also provided practical examples of the application of the platform to hotel data, demonstrating its potential to provide useful insights and improve hotel operations, for example, the platform was used to analyze customer reviews to identify common themes and areas for improvement.

Fitria et al. (Al Husaeni & Nandiyanto, 2022) provided a step-by-step analysis of bibliometric data using VOSViewer, which contributed in providing an easy way of data analysis by utilizing mapping tools. Bibliometric analysis was performed to produce network visualization of co-work maps and density maps of co-works. VOSViewer is used as a tool to visualize the analyzed data to be mapped. The authors demonstrate the use of VOSViewer for first-time users and, analyze research developments in digital learning media. This work involves performance of bibliometric analysis to produce network visualization of co-work maps and density maps of co-works. The authors collected data from Google Scholar using the Publish or Perish software application, which allows users to find the most cited authors, oldest and most recent publication years,

and bibliometric records of each research. The research focuses on analyzing digital learning articles in the pre- and post-COVID-19 pandemic period. The findings suggest that VOSViewer can provide suggestions in the data analysis results.

Omar et al. proposed a system (Omar & Jumaa, 2022) to improve big data analysis performance by using a fast big data processing engine using Apache Spark and a big data storage environment using Apache Hadoop. The system tests around 11 Gigabytes of text data from multiple sources for sentiment analysis and uses three different machine learning (ML) algorithms that are already supported by the Spark ML package. The system programs were written in Java and Scala programming languages, and the constructed model consists of the classification algorithms as well as the pre-processing steps in a figure of ML pipeline. The proposed system was implemented in both central and distributed data processing. The results showed that the system works efficiently for treating big data, it gains excellent accuracy with fast execution time, especially in the distributed data nodes.

In (Welo et al., 2023), this academic article delved into the significance of efficient data processing, analysis, and management for businesses in light of the exponential growth in data generation. The researchers Tembo et al. have primarily focused on exploring big data concepts and architectures, particularly through the use of the open-source Hadoop framework. The experimentation involves deploying Hadoop in a virtualized Linux environment with multiple nodes and comparing the processing time of the MapReduce algorithm across different datasets with varying numbers of data nodes. The work underscores the importance of a robust big data architecture for companies to effectively handle the immense volume of data generated. Furthermore, the research demonstrates that processing time decreases as the number of nodes in the cluster increases, suggesting significant potential for cost savings through big data processing.

1.3 Problem Statement

According to the different works available in of the literature, there are many gaps that need to be addressed. One of the most frequent issues that face developers when collecting data from particular websites is the standards involved and the approaches followed. In web data collection, developers face

challenges due to lacking standard frameworks. Traditional methods are time-consuming, prompting the need for frameworks integrating advanced techniques like Cloud Computing and parallel processing for faster data retrieval, efficient storage, and smooth handling of big data. Additionally, there's a shortage of curated datasets for Iraqi authors and universities, necessitating streamlined data retrieval and formatting for visualization and analysis.

Modeling biography data, particularly at scale, poses unique challenges. While existing literature suggests statistical or mathematical modeling, more sophisticated approaches are needed for deep analysis and visual exploration. Moreover, current metrics for academic performance often overlook collaboration aspects, highlighting the need for multidimensional metrics to accurately evaluate authors and universities.

In the light of the previous context mentioned above, universities are evaluated using standard metrics that reflect particular facts about a university such as the performance of its authors, education system, website, and other metrics that are currently used by many worldwide ranking agencies (e.g., webometrics, times, Shanghai, etc.). However, the literature includes many ranking methodologies which do not consider important aspects such as the level of collaboration among the universities or among the authors themselves. As a result, it is required to look at the performance of a university by deeply into the relations among authors and universities, which contributes to coming up with an accurate evaluation and eventually enabling officials to rank universities.

1.4 Objectives

This thesis aims to achieve the following objectives:

1. Propose standard measurement (Academic Capital) for assessing the academic performance of a university and author. These measurements are inspired by both educational platforms (e.g., Google Scholar) and network centrality measurements.
2. Constructing a web crawler that is capable of extracting biography data from Google Scholar, and establishing a connection to MongoDB Atlas to store the scraped data efficiently.

3. Leverage Cloud resources to deploy a scalable cluster architecture, integrated with Apache Spark to facilitate efficient parallel processing of data generated. By employing Spark SQL, the objective is to execute data processing operations, enabling rapid analysis and insights on the distributed database.
4. Implementing efficient data processing techniques by utilizing Apache Spark to process the collected biography data.
5. Enhancing data management capabilities through the implementation of MongoDB Atlas, a distributed database, to store, organize, and retrieve biography data both prior and after processing.
6. Storing the collected biography data in two files, one for the authors and the other one for the universities. Each file includes two portions: nodes portion and edges portion.

1.5 Contributions

The following four statements below represent the main contributions of this thesis:

1. A novel metric, termed as Academic Capital, was proposed for evaluating the academic performance of both; Iraqi universities and authors. The new metric uses biography information sourced from the collected data as well as network centrality metrics. Thus, it represents rank for Iraqi universities.
2. Proposed system architecture by integrating the concepts of parallel processing, Cloud Computing, big data, virtualization, and complex networks to manipulate Google scholar data.
3. Proposed a new approach is used to collect unique dataset that includes biography information of the published articles of the verified Iraqi authors in Google scholar.
4. Generate two special network model for both Iraqi authors and Iraqi universities, using concepts inspired by complex networks.

By combining Cloud, Big data, Parallel Computing, virtualization, Complex Network modeling, with proposed metric for performance evaluation, this thesis provides a comprehensive understanding of the academic capital dynamics of scientific research in Iraq. These contributions pave the way for evidence-based decision making, fostering collaboration, and accelerating the

growth of the scientific research sector, contributing ultimately to the advancement of knowledge and innovation in Iraq.

1.6 Thesis Layout

The remaining parts of the research are organized as following throughout the thesis:

Chapter 2: This chapter covers general background, where a general description of Academic Capital is provided. This is followed by the Web mining technology as well as the Web scraping architecture and related technologies and tools. After that, computing paradigms and big data which cover a large amount of big data and its frameworks, a general description of the cloud and its services, parallel processing, virtualization, and a general description of complex networks and their visualization platforms are discussed, respectively.

Chapter 3: This chapter provides an overview of the system tools used. The tools used for data scraping are described, alongside with their architecture. This is followed by explanations of the tools that are used to process big data, use of parallelism, and an understanding of the Apache Spark and MongoDB architecture in detail. The AWS cloud services are also explained, and then the complex networks and the measurements used for them are discussed, with a focus on co-authorship networks and collaboration networks. Finally, the platform used for network visualization has been addressed.

Chapter 4: This chapter describes the proposed system design and implementation. It explains Steps to implement the system sequentially, with an accurate description of the auxiliary tools and office packages and proposed algorithms for each step.

Chapter 5: This chapter assesses the results' discussion and analysis by taking samples during work experience.

Chapter 6: This chapter state the conclusion derived from this work, proposed future works, and challenges and problems.

CHAPTER TWO

General Background

2.1 Preface

This chapter delves into the concept of academic capital and also examines the use of web mining technology, which facilitates the growth of knowledge and personalized content. Furthermore, investigates the increasing role of big data processing in decision-making, with special focus on the cluster computing paradigm where networked computers collaborate to enhance performance and scalability in virtualization, parallel, and distributed computing. Additionally, a graphical representation of data, trends, and patterns is provided by complex network visualization tools, which are then presented. This chapter provides a comprehensive introduction to these fundamental concepts and underlines their significance in the advancement of academia and technology.

2.2 Academic Capital

Academic capital is a concept that was first proposed by Pierre Bourdieu in 1979. It refers to the potential that an individual possesses, based on their education and other academic experiences, to advance in society. Similar to other types of capital, such as social, economic, and cultural capital, academic capital is not dependent on a single factor. The word "academic capital" is used to refer to the accumulated academic achievements of a specific researcher, institution, university, or college (Lai & Shi, 2023).

Academic capital is a key element in the explanation of scientific productivity. However, few studies have analyzed this topic in the academic context, and their conclusions about composition and measurement remain ambiguous (Alba, 2022). Academic capital is highly crucial in the context of scientific research due to the fact that it can have a major impact on a person's capacity to obtain money, publish research, and advance in their career. This is especially true in the setting of universities. For instance, academics who have an extensive publication history and an established reputation have a greater chance of having their research ideas funded and of being invited to cooperate with other researchers on high-profile projects. In addition, educational institutions that have a stellar reputation are more likely to entice

outstanding researchers and students, both of which can contribute to a further increase in their academic capital (Mishra, 2020).

The overall concept of academic capital is one that is intricate and multi-faceted, and it is an essential component of scientific research as well as academic careers. By knowledge of the elements that contribute to academic capital being gained and efforts being made to eliminate gaps in access to these resources, better scientific advancement and discovery can be fostered, and a scientific community that is more egalitarian and welcoming for all individuals can be established (Bitzer & Matimbo, 2017).

2.2.1 Google Scholar

Large-scale sources of metadata about scientific publications have always been essential to the development of the bibliometrics field. In the end, bibliometricians employ these sources as the raw materials for their studies, so having access to them has been essential to the analyses in this field (Delgado López-Cózar et al., 2019). The resurgence and increase in interest in the scale of the academic web can be attributed to the introduction of academic search engines such as Google Scholar and Microsoft Academic Search. These academic search engines aim to index the entirety of contemporary academic knowledge. The GS is appropriate because it covers all subject areas and because the articles are classified, it is possible to do analyses based on the subject categories (Martín-Martín et al., 2018).

The degree to which academic articles or conference papers are seen by the public is directly proportional to how easy they can be located using academic search engines, most notably Google Scholar (Rovira et al., 2021). It not only has a greater retrospective coverage than WoS or Scopus, but it also includes a wider number of languages and publication kinds, and it is growing at a faster rate than either of those other databases. In addition to this, it is able to obtain a greater number of citations, the significance of which shifts depending on the subject matter being researched. Furthermore, the amount of time that students and researchers spend utilizing Google Scholar as their primary resource for exploring and making use of scientific material is continuously expanding (Martín-Martín et al., 2018) (Orduña-Malea et al., 2015). Since it does not place any limitations on the kind of papers or sources that can be searched, Google Scholar is an essential resource for bibliometric

research, which focuses on determining which scientific documents have had the greatest impact on their respective fields (Martín-Martín et al., 2017).

Coverage in Google Scholar not only grows when new documents are indexed, but also does two full recrawls of its index on an approximately yearly basis in order to determine whether or not certain documents have been removed from public access on the internet. If the crawlers used by Google Scholar are unable to access a document, that document is deleted from the index. There are some instances in which documents that have been removed are still retrievable via Google Scholar. This is either due to the fact that they are accessible via other sources on the Internet that Google Scholar also indexes, or because they are cited in other works that are indexed within the system (Martín-Martín & López-Cózar, 2021).

Google Scholar, in contrast to WoS and Scopus, uses an approach that is both inclusive and automated. It indexes any document that appears to be academic and that its crawlers can find and access on the web. This includes documents that are hidden behind paywalls due to agreements with their respective publishers (Martín-Martín et al., 2021). The secondary products are generated by Google Scholar and the availability of citation data. Since the data in Google Scholar offers a far more thorough understanding of the influence that publications have on their respective academic communities than the data in other citation databases, some bibliometricians and even scientists from other fields have become interested in Google Scholar Citations (GSC) and Google Scholar Metric (GSM). The conclusion has been arrived at that a more comprehensive perspective of the academic world is offered by GS as a result of the fact that a significant number of sources that were hitherto obscured from view have been shed light on (Delgado López-Cózar et al., 2019).

Massive crawler-based search engines like Google Scholar and Microsoft Academic have begun to make vast quantities of scholarly data easily accessible to everybody and everyone free of charge. Due to its accessibility and the ease with which its users are already acquainted with its search interface, Google Scholar has quickly become the most popular information resource in academic circles. Despite the fact that not all documents could be read in their entirety, Google Scholar was able to compile a sizeable

repository of documents that were open to the public and covered a wide variety of fields of study and languages (Gusenbauer, 2019).

2.2.2 Bibliographic properties of GS as a database

Empirical facts regarding GS's bibliographic characteristics as a database is aimed for in this section. Three factors, namely growth pace, coverage, and size, will be discussed (Delgado López-Cózar et al., 2019).

A: Size

One key characteristic that distinguishes GS as a big data source is its size. A search engine called GS displays the content that is currently available on the academic website., unlike Scopus or WoS, which are carefully restricted databases. The web is dynamic, unstable, and unmanageable. Thus, stability issues affect search engine size estimation methods.

B. Coverage

There are different types of coverage provided by Google Scholar as a data source for research assessment. The following is a summary of each type of coverage:

- ❖ **Source Coverage:** Google Scholar covers a wide range of scholarly sources, including books, thesis, conference, and papers. It also indexes content from both open access and subscription-based sources.
- ❖ **Geographic Coverage:** Google Scholar is a global database, indexing scholarly content from all over the world. However, the article notes that the coverage of certain regions, such as Africa and Asia, may be less comprehensive.
- ❖ **Linguistic Coverage:** Google Scholar indexes content in many different languages, including non-English languages. However, the article notes that the coverage of non-English content may be less comprehensive.
- ❖ **Discipline Coverage:** Google Scholar covers a broad range of academic disciplines, including both natural and social sciences, as well as humanities and arts. However, the article notes that the coverage of certain disciplines, such as law and business, may be less comprehensive.

C. Growth rate

GS is a constantly updating database of source papers and citations. Longitudinal investigations of Nobel Prize winners and senior scholars and retroactive growth in citations to classic papers have investigated its growth pace. GS discovers more citations and indexes as well as fresh source content to existing papers approximately two months faster than Scopus. In a recent study, GS surpassed WoS, Scopus, and ResearchGate in detecting early citations to library and information science journal articles. GS continues to expand and index its enormous academic database efficiently. Figure 2.1 below illustrates that GS includes all knowledge regions and communities (educational, scientific, and professional).

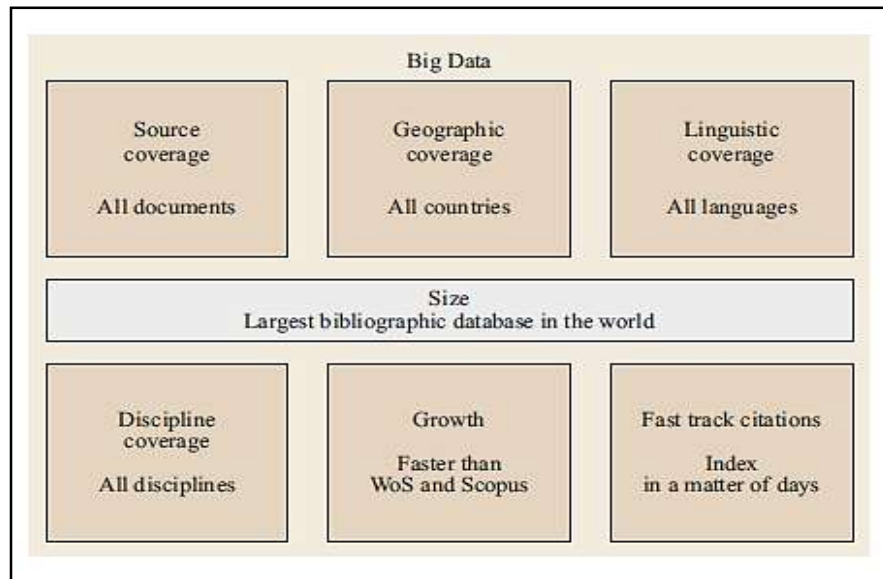


Figure 2.1 GS, a Big Data Bibliometric Tool (Delgado López-Cózar et al., 2019)

2.3 Information Retrieval

Data Mining can be considered as subfields of Information retrieval (IR) and information extraction (IE). Data Mining involves the process of discovering patterns, relationships, and insights from large datasets. Information retrieval, data mining, and information extraction are all intertwined in the process of web mining and are specific techniques used within Data Mining to extract, organize, and make sense of data. Information Retrieval and Information Extraction play crucial roles in identifying and extracting useful information from unstructured data, making it more amenable for analysis by data mining applications (Saurkar et al., 2018). IR is

an area of research which focuses on finding documents from a group of documents, whether relevant or not, by using keyword searches. With the rise of the internet, search engines have become the primary means of locating information online, making them a major part of IR. The technique of storing and retrieving information is referred to as IR, a term first coined in 1951 by Calvin Mooers. There are different types of IR, which are, traditional and automated. The latter two can be further categorized into different subtypes, as illustrated below in Figure 2.2. (Saini & Arora, 2016).

IE is a subset of artificial intelligence that focuses on extracting valuable information from unstructured data. IE systems typically aim to identify entities or objects, such as places, people, companies, and use extraction rules to extract relevant information. Examples of unstructured data include videos, images, audio, and text. Although the first IE systems primarily focus on text, it remains the most extensively researched type of data within both the academic and commercial communities. The ultimate objective of IE is to locate important components within unstructured data and extract them. so that through semantic classification, more relevant information may be produced. Several methods, such as web scraping, are employed to separate significant and valuable data from information that has been gathered (Saurkar et al., 2018).

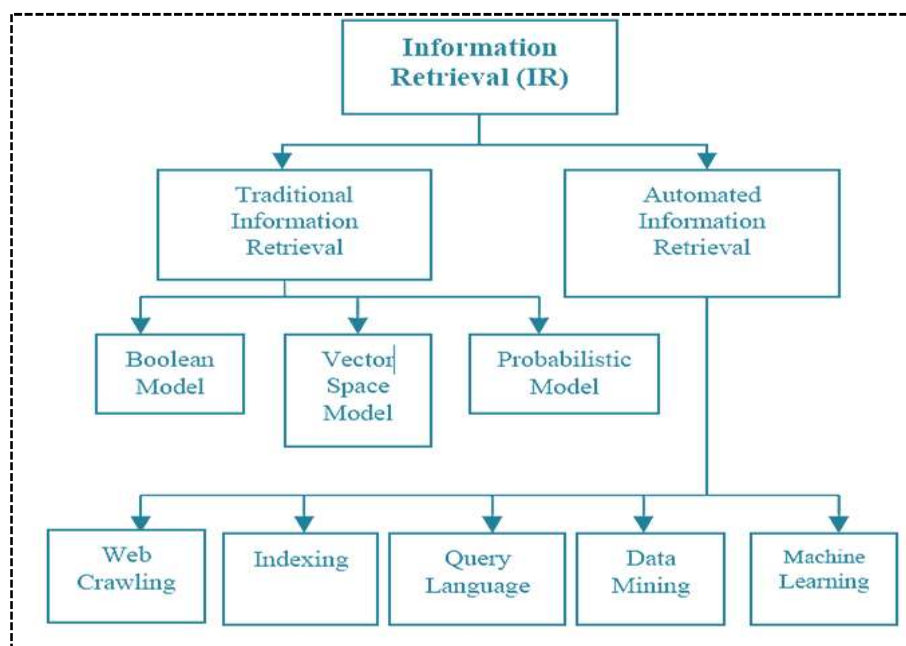


Figure 2.2 Methods of Information Retrieval approaches (Saini & Arora, 2016)

2.3.1 Web mining technology

Web mining is the process of gathering relevant and helpful data. It is used to create knowledge and customize content. Web mining is a type of program that gathers data from the web and extracts information from it by applying one or more architectural or use data. Web mining is the successor to data mining, and the two names mean the same thing. Data mining techniques are used in web mining to automatically retrieve and identify pertinent information from websites and web services. Web mining is a technique for identifying hidden relationships among vast volumes of data, which involves Web Crawler and Web Scraping (Sharma et al., 2022) (Kumar & Kumar, 2021).

Web scraping and web crawling are two techniques used to extract and analyze data from websites. These methods are popular among developers, businesses, and researchers, as they allow them to gather information for various purposes. Although both techniques are used to gather information from the internet, they differ in their approach and purpose. In the following two paragraphs below, the differences between web scraping and web crawling will be explored, and Table 2.1 following the two paragraphs will illustrate the comparison between them (Mitchell, 2018) (Al-Kerboly, 2023).

❖ Web Scraper

It is a program designed to extract data directly from a website's source code, including HTML, CSS, and other information. The scraped data is then parsed, giving the developer the ability to select which data, information, or text to extract. One of the challenges of web scraping is that there is no one-size-fits-all solution for all websites. It is currently not feasible to build a web scraper that works for every type of website. Instead, the developer must make adjustments to the code for each website they wish to scrape. This is because the scraper relies on specific website classes and names to collect and parse data (CARLE, 2020).

❖ Web Crawling

In contrast to web scraping, web crawling involves accessing web content and indexing it through hyperlinks. As a result, only the URL is extracted without any specific information. The full content can be accessed through the hyperlink, but it is usually not archived. Search engines such as Google, crawl

the web, analyze the online content, and compile all the links they find to match the search query. Crawlers, also known as spiders, are utilized in price comparison tools (Vanden Broucke & Baesens, 2018), (Mitchell, 2018).

Table 2.1 Comparison Between Web Scraping Vs Web Crawling (Al-Kerboly, 2023).

No.	Web Crawling	Web Scraping
1	Web crawlers index the content they discover by scanning links on webpages, generating a search index that search engines can use.	Web scrapers take information off of a website and organize it into a database or CSV file.
2	Often performed on a vast scale	feasible at any scale.
3	Deduplication is required, and the crawler listens to the robots.txt file.	Deduplication is optional. Scraper ignores the robots.txt file and views itself as a search engine.
4	Only requires a crawler or crawl agent	A crawler or crawl agent with a parser is required.
5	Data is saved by crawling.	Data is extracted by scraping.

2.3.2 Web Scraping

Given the current existence of approximately 1.2 billion websites as of May 2021, and over 4.5 billion internet users, this vast amount of available information can be challenging to locate and organize in a meaningful way (Andersson, 2021). Web scraping is a technique used to extract specific data from the HTML elements of web pages. This extracted data can be utilized for various applications such as product review analysis, weather data tracking, online price monitoring, etc. It allows for the collection of large amounts of data from the internet and involves retrieving web documents and extracting pre-defined information. The extracted information is then stored in a structured format for further use (Mitchell, 2018). Web scraping is a technique within the field of web mining that involves extracting information from multiple websites and consolidating it into a single database or spreadsheet. This allows for easy visualization and analysis of the data (Azhar et al., 2019). Web content is diverse and can be difficult to process due to various formats and access interfaces. This can make indexing and semantic processing challenging. Web scraping is a technique used to overcome these challenges by extracting specific data from websites, including social, financial, security,

and academic information. It is a useful technique for gathering data that may be difficult to obtain through other methods, particularly for research purposes (Sirisuriya, 2015). The overall goal of the web scraping process is to extract information from a website and transform it into an understandable structure like spreadsheets, database or CSV file as shown in Figure 2.3 below. (Sirisuriya, 2015), (Kisseljov, 2023).

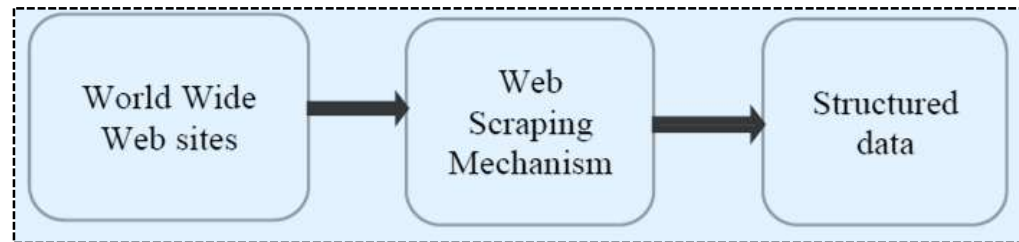


Figure 2.3 The Web Scraping Basic Architecture (Sirisuriya, 2015)

Creating a web scraper is different from utilizing a pre-existing API provided by websites such as Twitter and Wikipedia. For instance, the Twitter API allows users to access pre-organized, cleaned, and formatted data, which is more efficient than developing a web scraper for the same purpose. With an API, users can often select and modify the type of data they wish to collect and without proper request limiting policies. The reason behind this is the server being accessed may perceive large number of requests in a short time as a Denial-of-Service attack (Andersson, 2021). Various methods can be used for scraping data from a website, with web scraping standing as the most common. Web scraping can be achieved using different programming languages, with Python, Java, and PHP being among the most popular (Kisseljov, 2023).

2.3.3 Web Scraping Architecture

The three primary stages of web scraping are; website analysis, website crawling, and data organizing. These steps are interrelated. Web technology and computer language, such as R or Python are prerequisites for each phase as illustrated in figure 2.4 below, these stages frequently need some human involvement and cannot be entirely automated (Krotov et al., 2020), (Khder, 2021). The primary stages of Web Scraping are:

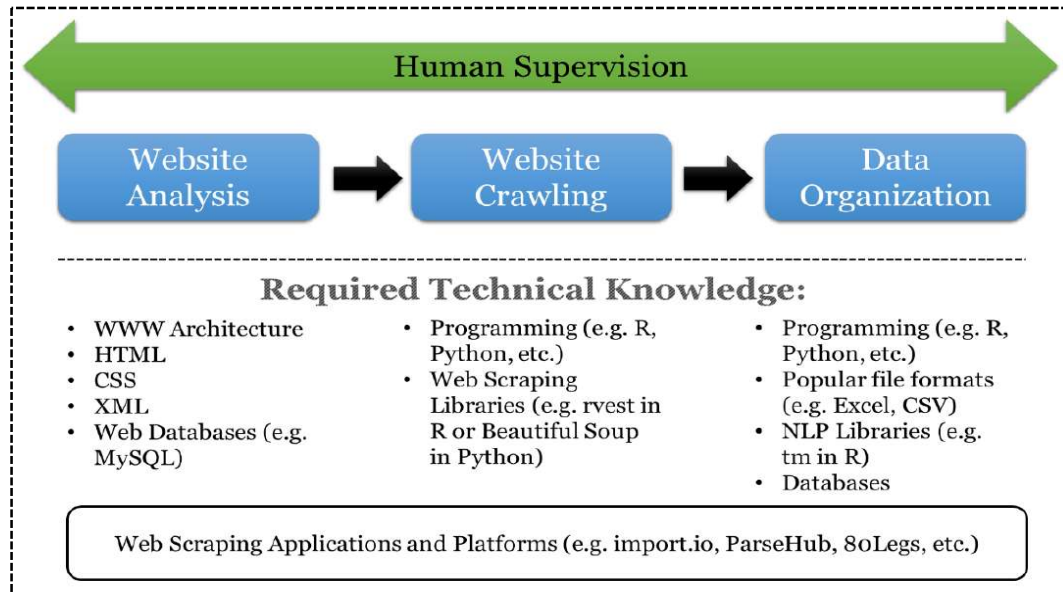


Figure 2.4 Web Scraping Phases (Krotov et al., 2020)

❖ Website Analysis

It entails looking at the fundamental design of a website or web repository, such as, an online database, to determine how the necessary data is kept. Understanding the architecture of the World Wide Web, markup languages like HTML, CSS, XML, and XBRL, and different web databases like MySQL are necessary.

❖ Website Crawling

It includes writing and executing a script to browse the website automatically and get the necessary information. Due to their frequent use in data science and the availability of libraries like Rvest and BeautifulSoup for automatic crawling and parsing of web data, these scripts are usually constructed using well-known programming languages like R and Python.

❖ Data Organization

It is used to facilitate additional analysis of the data. It needs to be cleaned, pre-processed, and arranged after being extracted from the chosen web repository. In order to save time, a programmed method is necessary due to the volume of data involved.

2.3.4 Web Scraping Techniques

According to, there are several techniques that can be used for web scraping, which are (Sirisuriya, 2015), (Meesters, 2021), (Khder, 2021):

- ❖ **HTTP Programming:** This technique involves retrieving data from both static and dynamic web pages, by sending HTTP requests to the remote web server using socket programming.
- ❖ **HTML Parsing:** This technique involves parsing the HTML code of a webpage to extract specific information. This can be done using libraries such as BeautifulSoup in Python.
- ❖ **DOM Parsing:** Document Object Model Programs can retrieve dynamic content created by client-side scripts through embedding a complete web browser. In order for programs to retrieve portions of the pages, these browser controls parse web pages into a DOM tree.
- ❖ **Web Scraping Software:** A variety of software tools are at one's disposal for tailoring web-scraping solutions. These tools can make an effort to identify a page's data structure automatically, or may offer a recording interface that eliminates the need to manually write web scraping code, some scripting functions for content extraction and transformation, and database interfaces for storing the data that has been scraped into local databases.
- ❖ **Computer vision web-page analyses:** Machine learning and computer vision techniques can be used to identify and get data from websites, by applying a human-like visual interpretation to the pages.
- ❖ **API Scraping:** Some websites provide APIs that allow developers to access data in a structured and organized way. This can act as a more reliable and efficient way to extract data compared to traditional web scraping techniques.
- ❖ **Machine Learning:** This involves training models to recognize patterns in the data and extract relevant information from unstructured web pages.

It is important to note that web scraping should be carried out ethically and in accordance with the website's terms of service. Additionally, measures may be in place on some websites to prevent web scraping. Therefore, it is important to be aware of any legal or ethical implications are considered before data extraction from a website.

2.3.5 Web Scraping tools

These tools can automate certain steps involved in the process and available as either Cloud-based platforms or desktop applications that can be used independently. Brief introductions to some of the prominent cloud-based web scraping tools, both free and paid software available are displayed below (Mitchell, 2018).

1. Webscraper.io

Webscraper.io is a web scraping tool that provides both a Google Chrome extension and a cloud-based service. The Google Chrome extension is free to use, while the cloud-based web scraper has a price depending on the number of URLs to scrape (Dale, 2022).

2. Scrapy

Scrapy is a Python-based, free and open-source framework that supports web scraping and crawling. It can also be used for data extraction using APIs. Scrapy Cloud provides scalability on demand. It offers both visual and code options, resource management, and full API access. However, current web scraping solutions focus solely on data acquisition, which is only the first step of big data (Dale, 2022), (Mitchell, 2018).

3. Selenium

Selenium is a tool that automates and emulates a human's browsing activity on a web browser. It is commonly used for testing web applications but can also be used for web scraping. By using the Selenium WebDriver API and a browser driver, such as Chrome Driver, Selenium can perform tasks in the same way a user would. This makes it possible to load and scrape web pages that use JavaScript to update the DOM. Selenium supports multiple programming languages, including Java, Python, C#, JavaScript, Haskell, and Ruby (Broucke & Baesens, 2018).

4. Import.io

This is a Cloud-based tool with a visual interface designed for extracting Web data that is suitable for use by individuals or organizations. It offers a Software-as-a-Service model, and all extracted data is stored on its cloud server, with the ability to download data in various formats as JSON, Excel, CSV, and Google Sheets, or access it via API. the Notable Features has a user-

friendly web-based interface that allows for easy setup of automated interactions with web forms and authentication (Saurkar et al., 2018), (Krotov et al., 2020).

5. Octaparse

This is an available tool that includes workflow automation and a Web interface for web scraping. The tool is considered mature and has undergone seven revisions. It features an intuitive point-and-click interface and supports web authentication. Additionally, it enables exporting data in multiple formats and supports scheduled crawling (Dale, 2022), (Krotov et al., 2020).

2.4 Computing Paradigms and Big Data

As big data has become more widespread in various fields, quick and efficient retrieval of large amounts of data has become a critical aspect of decision-making. Efficiently indexing and retrieving large datasets is a crucial metric for any indexing system (Elmeiligy et al., 2021). The advent of social media, internet, electronic transactions, and next-generation sequencing has not only created new possibilities but has also resulted in an immense volume of data (Khalid & Yousaf, 2021). The phrase "big data" does not only refer to the fact that data sets are extensive in size, but also to the degree of complexity and the rate at which they are gathered. The complexity of big data results in difficulty of processing it by using traditional data processing systems in a cost-effective and timely manner. This difficulty prompted the evolution of big data technologies, which assisted individuals and organizations to acquire, store, process, and analyze the data in a distributed environment (Chen et al., 2014).

The storing of data and its subsequent processing for analysis are both part of the big data processing. The conventional methods and tools were founded on the fundamental idea of a straightforward data structure; though, the vast majority of big data consists of unstructured data. Hence, in order to collect, organize, and interpret enormous data sets and get insight from them, big data necessitates the use of non-traditional technologies. Collecting, storing, and analyzing massive amounts of data, both structured and unstructured, is what is known as "big data processing," and it is done with the goal of gaining actionable insights and illuminating relevant patterns that might otherwise remain hidden (Rahman et al., 2021).

There are two different ways of addressing the issue of processing big data. The first method involves using a single computer that is capable of both extremely high levels of computing and storage. The second method involves constructing a cluster out of multiple conventional computers by interconnecting a large number of them. According to the definition provided by McKinsey & Company, the quantity of data in a dataset is not the sole factor that determines whether or not it constitutes big data. The two most important qualities are the scale of the data, which is always expanding, and the administration of that data, which cannot be handled by conventional database technology (Osman, 2019), (Chen et al., 2014). This rapid development of data leads to a number of critical issues, such as storage, security, scalability, and the extraction of usable knowledge, all of which are difficult to manage using normal methodologies, procedures, and tools. The data must first be comprehensible, then it must be analyzable, and finally, a conclusion must be reached from the data in order for it to have any value (Osman, 2019).

2.4.1 Big Data Architecture Components

Both Data Warehouses (DW) and Big Data (BD) share common components in their logical architectural abstraction, including data sources, (extraction, transformation, loading) (ETL), storage, processing, and analysis. However, DW typically relies on the organization's database systems as the primary data source, followed by flat files and older systems (Salinas & Lemus, 2017). Conversely, big data architectures often adopt a distributed approach rather than a centralized one, making them distinct from traditional data warehouses. Despite these differences, both DW and BD systems encompass key components such as storage, processing, and analysis systems within their architecture as shown in Figure 2.5. (Yadav et al., 2013).

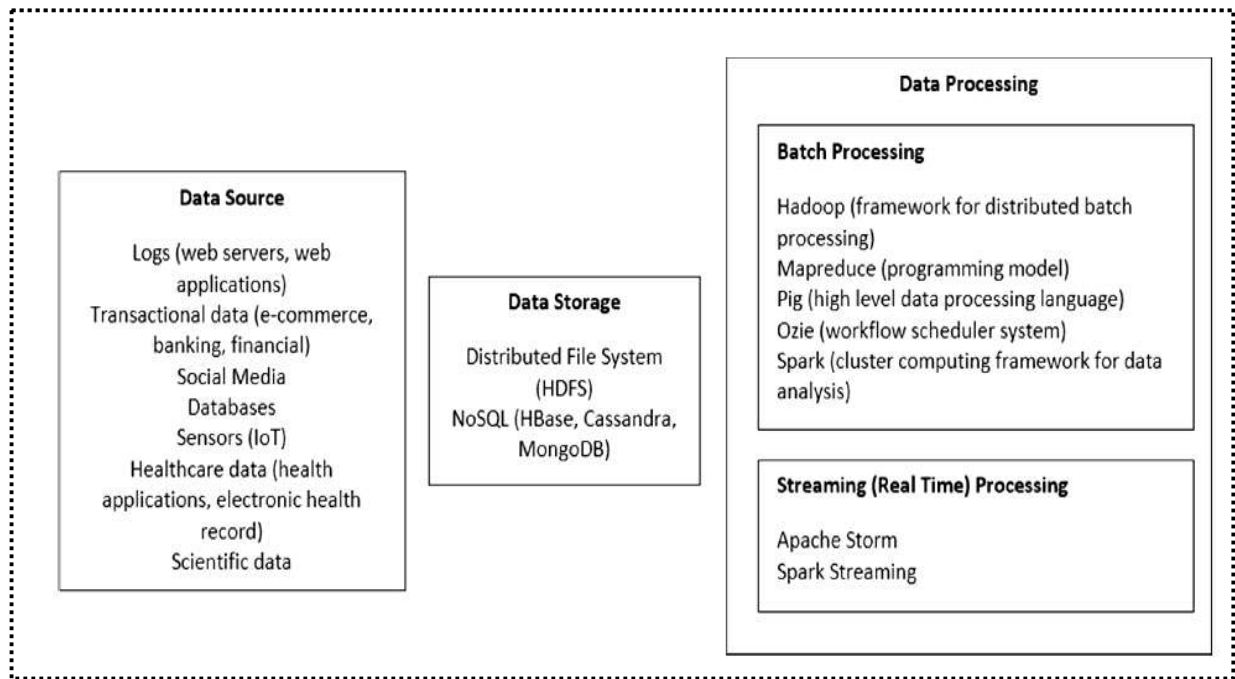


Figure 2.5 Numerous Big Data Architecture elements (Parlina et al., 2020)

2.4.2 Big Data Characteristics

Big data was originally described by Gartner using the phrases: volume, velocity, and variety, sometimes known as the 3Vs. Afterwards, veracity and value were added to the list of characteristics that define big data. Below, the Characteristics of big data are further elucidated: (Ramírez-Gallego et al., 2017) (Mittal et al., 2019) (Buyya et al., 2016).

- ❖ **Volume:** is a key characteristic used to describe big data. In general, the volume of data is a relative term, on this basis there is no fixed threshold for considering data as big data, however, the term big data is typically used to refer to large amounts of data that are difficult to store, manage, and process using traditional databases and data processing architectures, therefore specialized tools and frameworks are required for dealing with such data.
- ❖ **Velocity:** Refers to how fast the new data is generated and entered into the system as well as the speed at which the generated data must be processed. The rate at which information comes in from sources such as mobile phones and clickstreams, which are continually moving, shows the rate at which data is used to support interactions and is generated by interactions.

- ❖ **Veracity:** Is the capacity to ensure that data are accurate and reliable, which is essential for making informed decisions and deriving meaningful insights from the data. The goal of data quality measures is to ensure that the data is correct and consistent across all uses. This can be accomplished by the application of processes such as data cleansing, normalization, and de-duplication.
- ❖ **Value:** It refers to ability of data analysis to generate meaningful and actionable information that can contribute to decision-making, innovation, and competitive advantage for businesses or organizations.
- ❖ **Variety:** This characteristic refers to the variety of data sources, formats, and structures that cannot easily be captured and managed in a traditional database. Big data systems must be adaptable enough to handle captured data in whatever format it naturally exists, without attempting to impose a data model or structure, see Figure 2.6.

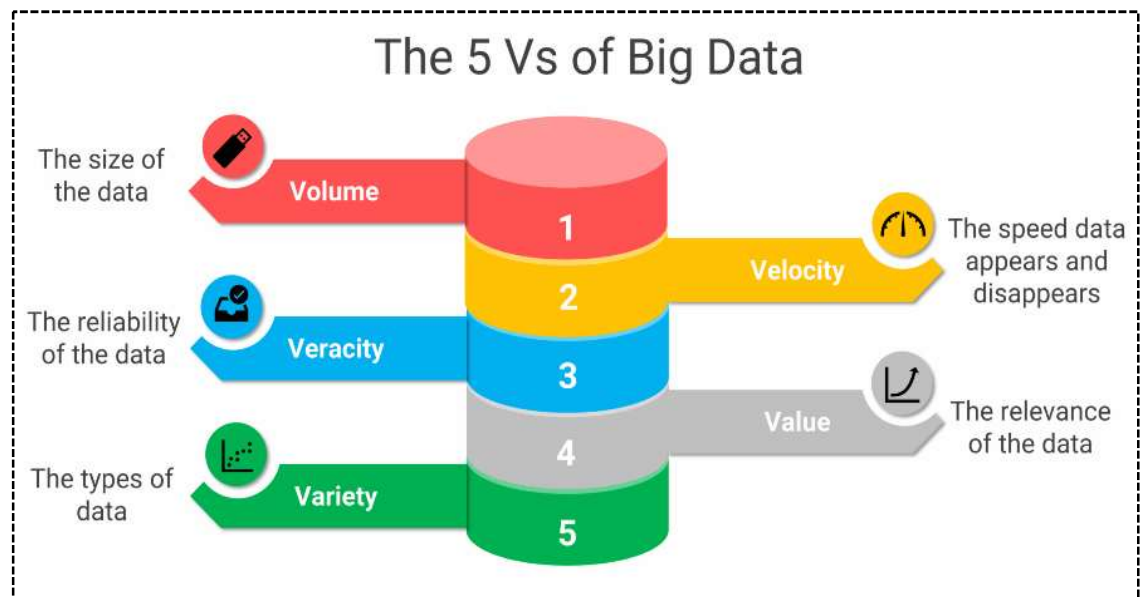


Figure 2.6 Characteristic of Big Data (Berisha et al., 2022)

2.4.3 Big Data Processing Frameworks

The division and distribution of data, as well as scalability, load balancing, fault tolerance, and high availability, are among the most important considerations. In order to give users high-level solutions and to provide an

abstraction layer over these functions, a variety of frameworks have been made available. The researches that have been done in this area has focused on several frameworks for big data, and the classification of some of these frameworks can be seen in Figure 2.7 which displays the prerequisites for the classification. Typically, these frameworks are categorized in accordance with the approach that they take to the processing of data. Batch, Streaming and hybrid processing frameworks are the three categories of large data processing frameworks. These frameworks are covered in the next subsections (Khalid & Yousaf, 2021).

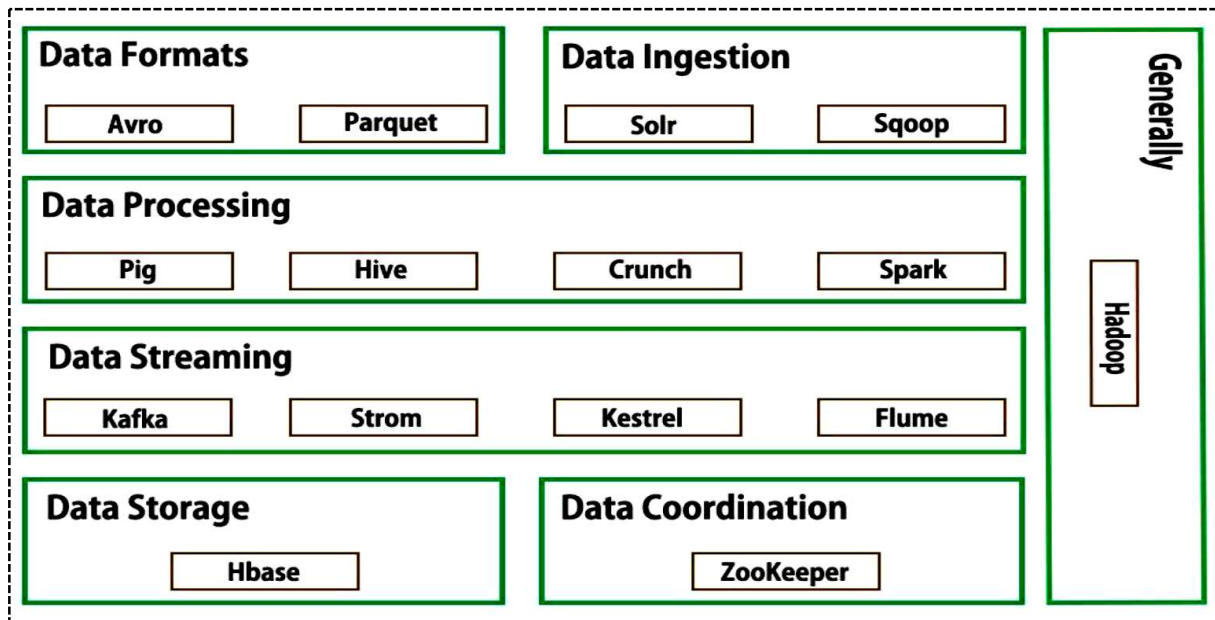


Figure 2.7 Big Data Framework (Khalid & Yousaf, 2021)

A. Frameworks for Streaming Processing

Streaming processing, which deals with continuous data, must be utilized in order to transform large amounts of data into data that can be processed quickly. It is necessary to feed data into an analytical tool, typically in the form of micro-batches and in real time. Apache Kafka, Apache Storm, and Apache Samza are the most popular frameworks for stream computation (Berisha et al., 2022).

B. Frameworks for Batch Processing

Batch processing is a sort of processing in which a number of logically related jobs run sequentially or in parallel, and the results of all of the individual jobs are combined to produce a final output. The gathering of data

for processing in batches, followed by processing of that data to produce an output that may then be utilized as an input for another process, is what is referred to as batch processing. It is suitable for applications that require a response, but do not place a crucial emphasis on response speed. Hadoop is the framework of batch processing that is most extensively utilized (Elmeiligy et al., 2021).

C. Frameworks for Hybrid Processing

Batch processing and stream processing are both manageable under hybrid processing systems. Spark and Flink, both developed by Apache, are examples of hybrid frameworks. Both frameworks due process in memory, making their data processing latencies significantly better than those of Hadoop (Khalid & Yousaf, 2021).

2.4.4 Cluster Computing

A type of parallel or distributed computing known as "cluster computing", which makes use of connected computers to operate as a single, integrated computing resource. In a cluster, various independent computers often referred to as nodes cooperate to carry out tasks and process data, providing superior performance and scalability than conventional computing methods (Kumari et al., 2021).

2.4.4.1 Cluster Computing Architecture

Integrated system is created by the cooperation of various key elements in a cluster computing design. These elements consist of (Hwang et al., 2013) (Zomaya, 2006):

- 1. Nodes:** Processing nodes can be any number of separate standalone commodity workstations, personal computers, or servers.
- 2. Interconnection Network:** In order to promote communication among the nodes that make up the cluster, a network must be established. To facilitate effective data sharing within the cluster, high-speed networking technologies like Ethernet or InfiniBand are frequently used.
- 3. Middleware:** By decomposing the job into tasks and distributing them to the worker nodes, the cluster operating system also serves as middleware. The

operating system of the nodes should be able to run applications that support a large number of users, numerous processes, and threads.

4. **Cluster Management Software:** The tools and interfaces required to administer and operate the cluster are provided by cluster management software. It enables administrators to handle job submission and management, resource allocation, task scheduling, and cluster health monitoring.

5. **Distributed File System (DFS):** The cluster's data is stored and managed via a distributed file system. Data access and sharing amongst compute nodes are made possible by the unified view of the file system it gives across numerous nodes.

6. **Resource Manager:** The resource manager is in charge of allocating and overseeing the cluster's computing resources. By organizing tasks and optimizing resource usage, it assures effective use of resources like CPU cores, memory, and storage. A typical system is shown by the cluster computing architecture in Figure 2.8

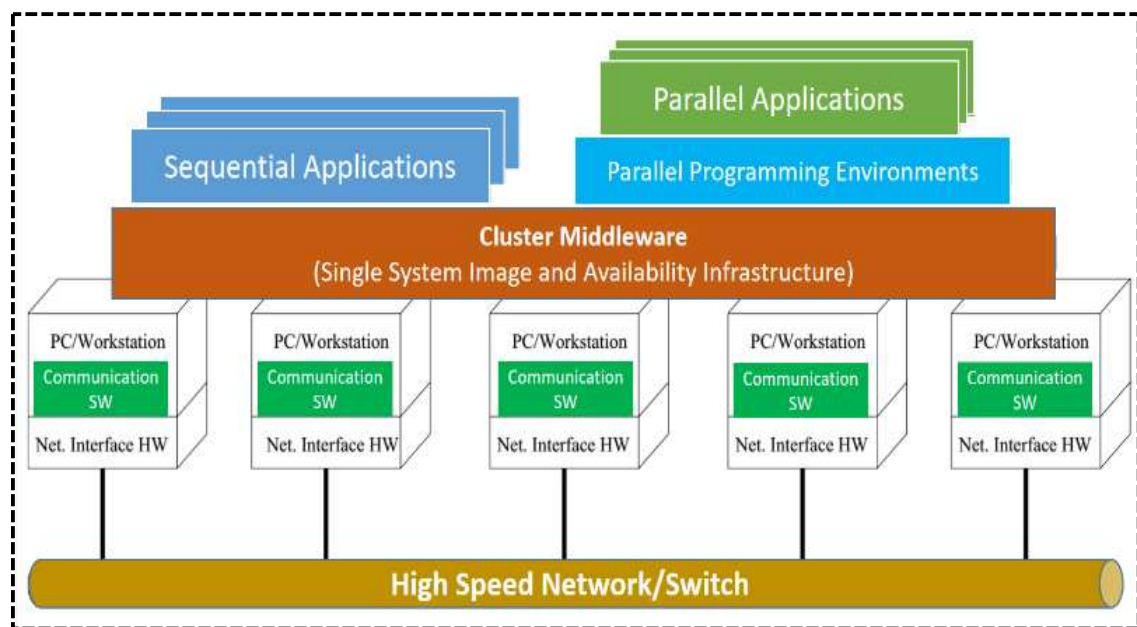


Figure 2.8 Cluster Computing Architecture (Hwang et al., 2013)

2.4.4.2 Cluster Computing Types

Clusters can perform a wide range of tasks due to their configuration, including the control of computationally demanding workloads and web-based applications. Clusters can be divided into two different types based on their functions (Balusamy et al., 2021).

- **Load Balancer:** This component equally distributes the computational effort among the cluster's compute nodes. In order to accomplish effective processing and prevent bottlenecks, it helps optimize resource utilization and ensures that tasks are spread equitably.
- **High availability:** This type of cluster used to guarantee continuous operation and reduce downtime. Redundancy and fault-tolerant designs are frequently used in these systems to deal with node failures and sustain continuous cluster functionality.

2.4.4.3 Cluster Computing and Big-Data Analytic Platforms

Cluster computing, originally proposed in the 1960s, involves multiple computers working together to solve complex problems. Early clusters were user-built for specific applications. In the 1980s, commercially successful general-purpose computer clusters like DEC's VAXcluster emerged. The VAXcluster system, introduced 40 years ago, featured closely-coupled computers with separate processors and memories connected via message passing. To overcome vendor-specific limitations, solutions like Parallel Virtual Machine (PVM) and Message Passing Interface (MPI) were developed in the 1990s. PVM provided a runtime environment for message passing, task management, and fault notification. MPI, a standard defining library routines, combined useful features from various systems. MPI lacked fault tolerance and was suitable only for controlled, reliable environments. As clusters grew, building and maintaining high-availability clusters became costly and complex (Kronenberg et al., 1986).

The fast rise of digital data, which was generated from a variety of sources over the past decade, has rendered traditional techniques of storage, processing, and analysis useless. As a response, new cluster computing frameworks have been developed to process and store very large datasets. As a direct consequence of this, a number of different execution frameworks for processing big data have surfaced. Hadoop MapReduce, a pioneering

framework, cleared the way for future frameworks that increase the processing and development of large-scale data in a variety of ways, and it did so by paving the way for future frameworks. The openness of the frameworks for big data cluster computing makes it simpler to develop large-scale data analysis applications that are capable of running on commodity hardware that is relatively inexpensive. Cloud computing, which clusters multiple computers together through the use of virtual machines, has been a driving force behind the increased usage of these newly developed data processing frameworks. One further essential aspect that has significantly contributed to the rise in popularity of modern cluster computing systems is the fact that these systems typically contain a high-level programming interface that simplifies the process of application creation (Khalid & Yousaf, 2021).

2.5 Cloud Computing

The technology of cloud computing has undergone rapid development in all countries over the course of the past few years. Recently, many of the cloud technologies are highly utilized by both individuals and organizations in a variety of fields, including broadcasting, marketing, care health, education, and others. Several sectors are putting money into cloud computing in the hopes of improving the effectiveness of their company operations, by reducing the amount of time, money, and labor required (Al-Khayat et al., 2016). A data processing system that is based at top of several servers, and which has the capacity to store and process enormous amounts of data is referred to as a big data system. The technologies of the cloud and big data are increasingly converging, which promises more efficient and cost-effective delivery of big data services. Big data is essential for many applications due to its capacity of providing high-value information and knowledge. Some examples of these applications include decision assistance and forecasting. The technologies of cloud computing and big data are now merging to make it possible for businesses to outsource their data to the cloud and derive value from it. Big data systems frequently make use of computer clusters as a means of improving their scalability and achieving better cost-performance ratios, Figure 2.9 illustrates simple architecture of cloud computing (Liu et al., 2018).

Cloud computing is a model that allows network access to a shared pool of reconfigurable computing resources (servers, networks, storage, apps, and services) to be available anywhere, anytime, and conveniently. With little administration work or service provider contact, these resources can be quickly provisioned and released. Cloud computing is a model which prioritizes availability, and it is made up of some fundamental features, three service models, and four deployment methods. All of these elements work together to form the model (Ruparelia, 2016).

The processing of big data can also be supported by cloud computing, which offers powerful storage, compute, and distributed capabilities. It is necessary to implement new data analysis algorithms as well as new methodologies in order to successfully manage the exponential increase in data volume and the requirements of massive-scale analytics in order to realize the full potential of big data. For this reason, the economies of scale are one of the fundamental benefits that comes along with providing services over the cloud. When a service provider makes use of the cloud's underlying infrastructure, it is possible to provide services that are superior in quality, while also being more affordable and dependable (Hill et al., 2015).

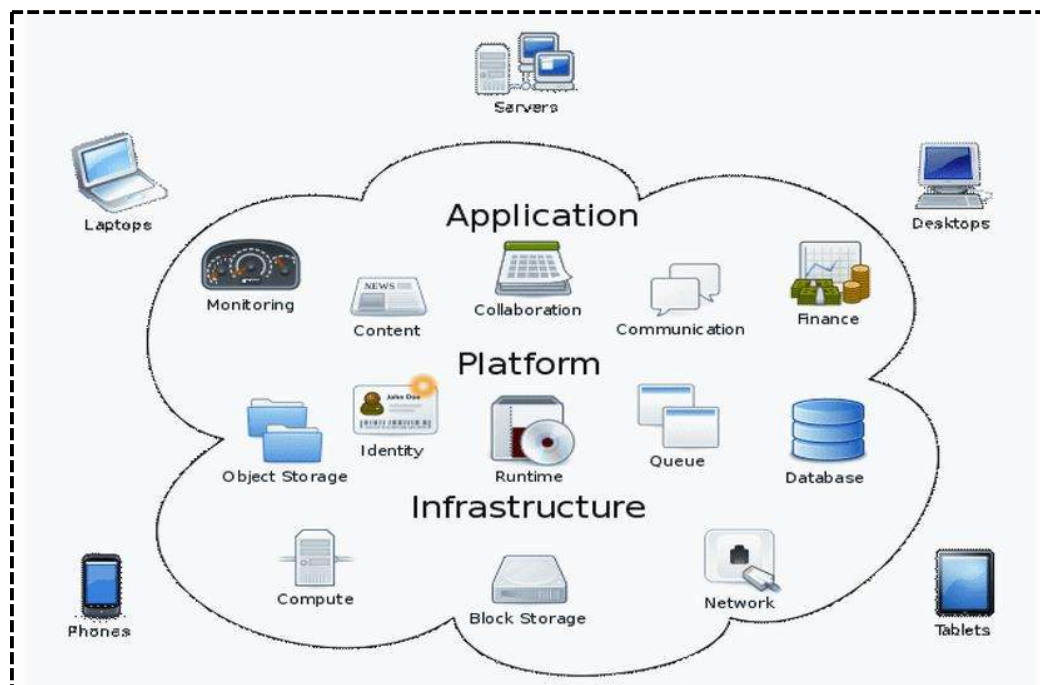


Figure 2.9 Simple Architecture of Cloud Computing (Liu et al., 2018)

2.5.1 Cloud Computing Layers

The cloud computing platform offers its services in accordance with three primary models. To elaborate and make things more apparent, a cloud computing platform comprises storage, network, and processing, with the cloud architecture being built of the following three layers (Al-Khayat et al., 2016; Malik et al., 2018; Srivastava & Khan, 2018):

- ❖ **Software as a Service (SaaS):** Is a cloud computing service model where software applications are hosted by a third-party provider and made available to customers over the internet. In this model, users access the software through a web browser or application interface, eliminating the need for installation and maintenance of the software on individual devices. In addition, the customer does not have any ability to govern or administer the cloud infrastructure. Facebook, Office 365, and Google Gmail are all examples of models that use the SaaS delivery method. SaaS provide the highest level of abstraction.
- ❖ **Platform as a Service (PaaS):** Is a cloud computing service model that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the underlying infrastructure. In PaaS, providers offer a comprehensive platform, including development tools, database management systems, middleware, and runtime environments, over the internet. Microsoft Azure, Amazon Web Services (AWS), and Heroku are all examples of platforms available through the PaaS approach. PaaS is more complicated in use than SaaS but it is easier than IaaS.
- ❖ **Infrastructure as a service (IaaS):** In IaaS, customers rent virtualized infrastructure components such as virtual machines, storage, and networking resources from a cloud service provider, rather than purchasing and maintaining physical hardware. In IaaS, user is not responsible for handling the cloud infrastructure, but they have type of controlling over applications, operating systems, and storage, and may have little control over selected networking components like firewalls. Amazon Elastic Compute Cloud (EC2), Rackspace, and Google Compute Engine are a few examples of IaaS. These three fundamental tiers of cloud computing are illustrated in Figure 2.10.

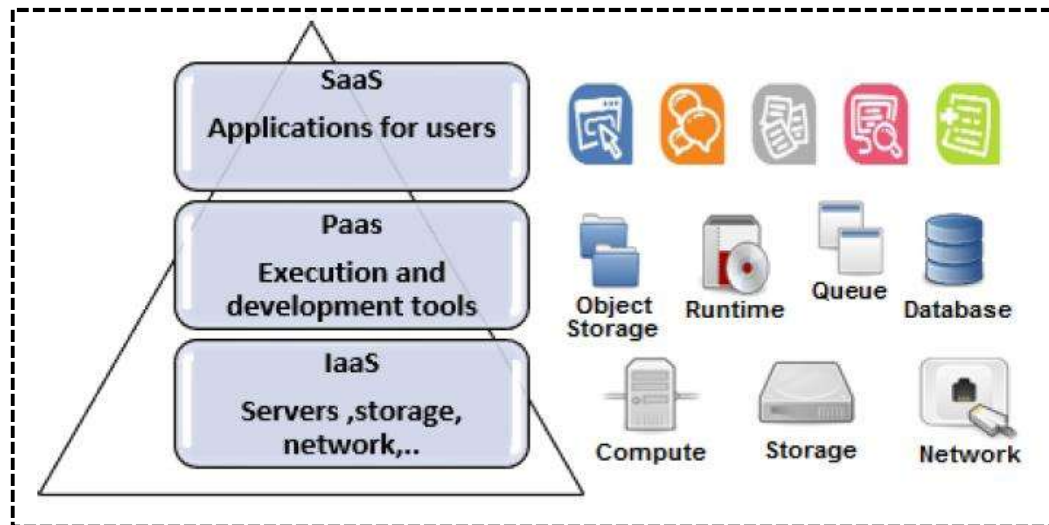


Figure 2.10 Cloud Computing Services (Malik et al., 2018)

2.5.2 Cloud Computing Deployment Models

Cloud computing allows developers the ability to focus on what is important. The proliferation of cloud computing has led to the development of a variety of models and approaches to cater to the requirements of individual users. There are varying degrees of command and control available with each variant of these services. The following is a list of different approaches that can be used to make use of cloud computing models: (Fox & Hao, 2017; Karataş & Akbulut, 2018)

- ❖ **Private Cloud:** Cloud providers cannot be hosted from the outside for security reasons. It is typically carried out either by the organization itself or by another business operating independently. Since the Private Cloud is a structure that is built by users themselves, they have complete control over the information stored there.
- ❖ **Public cloud:** refers to cloud computing services that are made available to the general public and are typically owned by large corporations. These services are either completely free or charged on a pay-per-use basis. In most cases, public cloud service providers like Microsoft, Google, and others manage their own infrastructure resources, and access is only made available through the Internet.

- ❖ **Community Cloud:** A model of cloud computing in which the underlying hardware and software are made available to several users from the same company. Community members get access to applications and data.
- ❖ **Hybrid Cloud:** Mixture of several or all of the cloud service paradigms. The goal is to mix several cloud services in order to produce a computer environment that is effectively controlled and highly automated. Within this category of cloud, the owner has the ability to move both data and applications between public and private clouds.

Table 2.2 below presents a comparison of service models available through cloud computing (Hill et al., 2015)

Table 2.2 Comparison of Cloud Computing Model (Hill et al., 2015)

Model	Scope	Managed by	Security Level
Public Model	General public and industries	Cloud service providers	Medium
Private Model	Single organization	Single organization	High
Community Model	Organizations having similar policies and same security concerns	Many organizations or cloud service providers	High
Hybrid Model	Public and organization	Public and organization	Medium

2.5.3 Cloud Characteristics

The following are some of the features that cloud computing should have in order to fulfill the requirements of customers and users and to deliver high-quality services (Fox & Hao, 2017) (Malik et al., 2018).

1. **Maintenance:** Cloud-hosted apps eliminate the need for individual installations, reducing costs and simplifying maintenance.
2. **High availability and reliability:** Cloud infrastructure offers greater reliability, minimizing the risk of infrastructure issues.
3. **High scalability:** Cloud services can rapidly scale resources to meet demand without manual coordination.

4. **Multi-sharing:** Distributed computing enables multiple users and applications to share infrastructure efficiently, reducing costs.
5. **Cost-efficiency:** Cloud eliminates the need for setting up and maintaining individual infrastructure, with payment based on resource usage.
6. **Resource sharing:** Cloud enables multiple users to collaborate on shared resources, including databases and hardware infrastructure.
7. **Measured service:** Cloud providers monitor and manage all aspects of the service, including capacity planning, billing, and optimization.
8. **On-demand self-service:** Users can access cloud services and resources anytime and cloud providers offer APIs for accessing services, with customers paying based on usage.
9. **Broad network access:** Cloud resources are accessible over the internet from various devices, providing flexibility and global accessibility.

2.6 Parallel Processing

Due to the ability of using several computing resources at once to solve a calculation problem, parallel computing is significant, in respect to the issue of being divided into discrete components that can be resolved simultaneously or in parallel. Having access to additional memory or getting better speed are the two main benefits of using a parallel computer. Since the total memory is the sum of the individual memories, the memory gain is simple to describe (Vázquez Benítez, 2015).

Large-scale data analysis is typically difficult, time-consuming, and computationally intensive. These data must be processed quickly due to the high rate at which they are generated. It also needs to be examined and processed effectively, which calls for the appropriate tools and methods. This procedure can be streamlined with Apache Spark and Apache Hadoop through parallel and distributed processing, and the primary purpose of them is to facilitate the collection, storage, and processing of increasing volumes, velocities, and varieties of data (Sivarajah et al., 2017).

In parallel processing, combining two or more computers to solve a single problem is common. The allocation of the processing work among several active processor cores, is a prerequisite for parallel processing. In order to accomplish a single goal, parallel processing uses a collection of

processes that run concurrently and communicate among themselves. The goal of parallel processing is to efficiently run the programs since doing so enables applications to run more quickly and provides concurrency, which is useful when carrying out several activities at once. It also makes better resource management possible. The use of many computers, which allows for the solution of larger and more complicated issues, can quickly alleviate the issue of a single computer's limited memory capacity (Chanda, 2021). Web companies such as Facebook, Google, Microsoft, and Yahoo! as well as open-source software foundation's such as Apache have designed big data frameworks, which can highly parallelize data processing based on data partitioning, that is using data-based parallelism (Liu et al., 2018).

2.7 Virtualization Concepts

With no doubt, computing is becoming more complex over time and in need of larger infrastructure. If more processing power is required, the corporations make significant investments in the acquisition of new physical infrastructure. Even with more infrastructure purchases, there was relatively little use of processing power and little return on investment. This encouraged businesses to use virtualization technology to optimize ROI and resource use. Before virtualization, one operating system and its applications ran on a single computing infrastructure, which led to underutilization of resources. In order to fulfill their growing computing needs, businesses had to buy new hardware resources because the hardware infrastructure was not shared. Following the introduction of virtualization, many operating systems and applications may use a single computing architecture. Through virtualization, customers can be assigned computer infrastructure according to their actual demands, making better use of the resources and saving significant amount of money on the expensive purchase of new hardware (Tavakkoli, 2019).

Virtualization is the term for the technology that allows several systems to be deployed within a single computing architecture. Virtual Machines (VMs) are separate execution environments that are separated from a physical computing system and have the appearance of individual computers (Ahmadian et al., 2018). Software operating on virtual machines typically does not have access to the implicit hardware resources of the host machine. Through the use of hardware-assisted virtualization, virtualization

performance can be enhanced by enabling virtualization in hardware, such as CPUs and other components. A virtual machine in platform virtualization is also referred to as a guest, and the software (firmware) responsible for creating virtual machines on the host Virtual Machine Manager (VMM) is known as a hypervisor, or virtual machine monitor (Ganesh et al., 2022).

A unique type of software named as hypervisor, is used in virtualization to enable the creation and operation of several virtual machines (VMs) on a single computing system. The hypervisor operates as a platform for the virtual machines (VMs) and communicates directly with the real computer system's resources. Generally, one type of virtualization technology, that uses thorough hardware emulation and an entire operating system installed on top of it, is called hypervisor-based virtualization (Abuabdo & Al-Sharif, 2019). Software-created computer systems are represented by virtual machines (VMs). Similar to a real computer, a virtual machine (VM) includes its own virtual CPU, RAM, hard drive, and network interface card (Turban et al., 2018). the Hypervisor layer of software is responsible for creating virtual machines (VMs). Hypervisor allows many operating systems to use a single computational infrastructure. In this instance, it seems that the host's physical resources are allotted to each OS. Hosted hypervisors run on top of the operating system of a machine. The OSs running in the virtual machines (VMs) are referred to as guest operating systems, and the native OS is known as the host operating system. Hosted hypervisors like QEMU, VMWare, and VirtualBox are examples of such. In Figure 2.11, the architecture for this method is shown (Ahmadian et al., 2018).

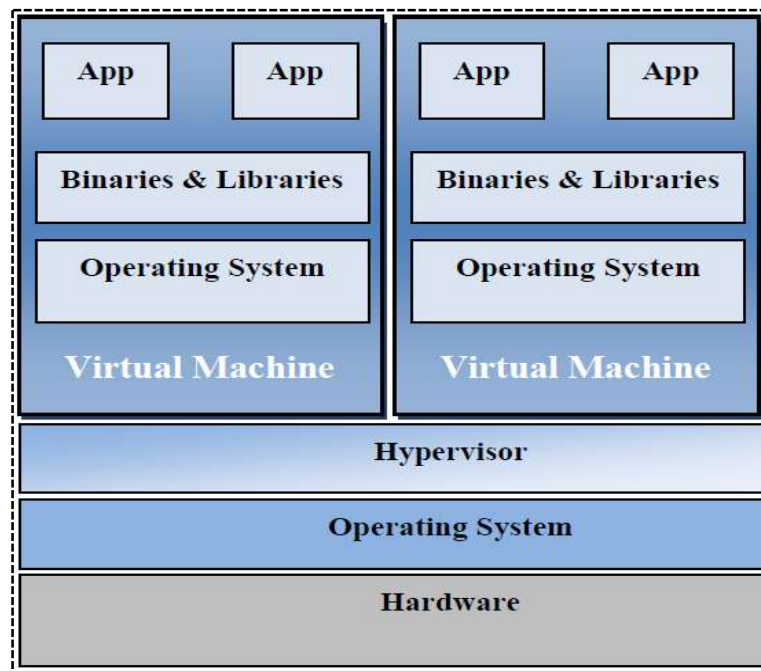


Figure 2.11 Hypervisor-based Virtualization Architectures (Ahmadian et al., 2018)

2.8 Complex Network and Graph Theory

According to the theory that underpins the concept of network theory, Complex Networks are graphs (or networks) that differ significantly from regular graphs or networks that are uniformly random. Simple networks, however, do not typically have many of the characteristics that are present in complex networks. These characteristics are typically seen in more complex networks. The study of complex networks was initiated at the beginning of the twenty first century, deriving from the concept of graph theory and being motivated by empirical examination of real networks such as biological networks, brain networks, social networks, and technology networks (Lima et al., 2018).

In the most recent decade, both the theory and implementation of complex networks have made remarkable strides and have become a topic of significant interest. The primary reason for this is the enormous number of tools that have been developed for the purpose of assessing various topologies of a great variety of real-world systems. These tools also provide the flexibility and generality which are necessary to accurately describe any natural structure (Rossetti & Cazabet, 2018). One way to see a complicated network is as a vast collection of vertices and edges that are interconnected

with one another. A computer, a person, an organization, or even a biological cell can all be represented by the concept of a vertex in graph theory. In this context, "interconnected" refers to the possibility of establishing a link between two vertices, such as when two businesses trade goods or when two individuals are familiar with one another (Kartun-Giles & Bianconi, 2019).

2.9 Complex Network Visualization Platforms

It is possible to describe data visualization as the process of displaying data using graphical representations such as graphs, charts, and pictorial forms. A data visualization tool allows us to make sense of data, trends, and patterns in data. It is possible for decision-makers to see analytics in a visual form by data visualization, which also makes it possible for decision-making and the determination of courses of action depending on what is seen in the visualization. People operating in this field make use of a wide variety of platforms for Complex Network (CN). It is possible for them to be high-level programming languages like Python, or more specialized platforms with plugins and apps, like Gephi, Cytoscape, and NetMiner (Sadiku et al., 2016). Some of the most widely used complex network visualization platforms are:

2.9.1 Cytoscape

Cytoscape is a popular and powerful open-source software platform used for visualizing, analyzing, and modeling complex networks. Cytoscape allows researchers and analysts to create, manipulate, and visualize networks, making it a valuable tool for understanding the structure and behavior of various types of networks, including biological, social, and technological networks. Despite being created initially for biological research, Cytoscape, now adays, is a common platform for sophisticated network analysis and visualization. A foundational set of features for data integration, analysis, and visualization are offered by the Cytoscape core distribution (Doncheva et al., 2018).

2.9.2 NetMiner

A Network Forensic Analysis Tool (NFAT) for Windows is called NetworkMiner. NetworkMiner is used as a passive network sniffer or packet capturing program to find OS systems, sessions, hostnames, open ports, and other information without creating network traffic. NetworkMiner can also

regenerate/reassemble transmitted data as well as certificates, for offline examination (Rattanaavayakorn & Premchaiswadi, 2015).

2.9.3 Gephi

Gephi is a dedicated open-source software tool for network analysis and visualization. It provides an interactive and user-friendly interface for exploring and visualizing complex networks, making it suitable for studying social networks, transportation networks, and other network-based data (Fonseca et al., 2016).

2.9.4 Pajek

Pajek is a software package for the analysis and visualization of massive networks (networks with up to one billion vertices, where there is no upper restriction on the number of lines except for memory size), which has been accessible for twenty years. A desktop-based platform for network data visualization and analysis is offered by Pajek. Both commercial and noncommercial versions are available. Pajek enables designs like mapping that is based on the Visualization of Similarity (VOS). In the drawing window, it can apply styling characteristics like vertices and edge colors and scaling. Pajek offers network transformations, cluster and community identification, different centrality, degree, and coefficient metrics calculations (Rattanaavayakorn & Premchaiswadi, 2015).

CHAPTER THREE

System Frameworks and Tools

3.1 Preface

This chapter presents the theoretical background and discusses several topics that form the cornerstone of this dissertation. To efficiently harness the power of big data, organizations use modern methods and technologies, such as, distributed computing, parallel processing, cloud computing, data mining. Working with big data, on the other hand raises many challenges, including data warehousing, data integrity, data quality, data privacy and security risks. All these subjects are used in order to create the academic capital of Iraqi authors and universities, and to show the patterns of cooperation between Iraqi authors. This chapter describes all the tools used in the system. Since choosing the right tools is at the heart of this toolkit. Following some brief explanations covered in the previous chapter, the sections in this chapter will provide detail and tool selection in construction

3.2 SerpApi

SerpApi is a cloud-based API that allows developers to scrape search engine Results Pages (SERPs) from Google, Bing, Yahoo, and other search engines. A scraper is a software program that extracts data from websites, and a Serp scraper specifically extracts data from search engine results pages. One of the key benefits of using SerpApi for scraping search engine results pages is to handle a lot of the complexity of web scraping automatically. For example, SerpApi handles issues such as changes to the structure of search results pages. This allows developers to focus on building their own applications rather than dealing with the intricacies of web scraping (valavandan et al., 2022).

Scraper using SerpApi works by making an HTTP request to the SerpApi endpoint with search parameters, receiving a JSON response containing search engine results data, and using that data in their own application (Johnson et al., 2022). The architecture of SerpApi is based on a cloud infrastructure that distributes requests to different servers in real-time, ensuring high availability and reliability of the service. Whenever a user sends a request to SerpApi, it is first processed by the load balancer, which selects the best server to handle the

request based on factors like server load, network latency, and geographical location. The selected server then executes the search query on the relevant search engine, scrapes the relevant information from the SERP, and delivers the results to the user in a structured format (Luo et al., 2020).

SerpApi does not allow for an unlimited number of free usages. Since it is limited to only 100 free Google searches each month, testing capabilities are severely restricted (Johnson et al., 2022).

The architecture of SerpApi is one that is based on microservices. This indicates that the functionality of the tool has been separated into individual services that are capable of communicating with one another through the use of APIs. Each microservice is tasked with performing a unique function, such as interpreting search results or managing API calls, and may be expanded individually to accommodate shifting requirements. The traditional monolithic architecture has a few drawbacks, but the microservices architecture has some advantages over it, including enhanced scalability, increased flexibility, and increased reliability. Changes and modifications can be made to specific services within a system that utilizes microservices architecture without having an impact on the system as a whole. In addition, the failure of a single service does not necessarily bring down the entire system, which contributes to the system's resilience. Since it is built on microservices architecture, SerpApi is able to efficiently manage high volumes of search queries and offer users with results that are both quick and accurate. The application is developed on a platform that is hosted in the cloud, and its services are distributed across a wide variety of servers and data centers to achieve both high availability and fault tolerance (Sultan & Abdullah, 2022).

3.2.1 Design of SerpApi Strategy

To ensure seamless interaction with the SerpApi API and deliver accurate search results efficiently, the following strategy is employed:

1. The user initiates communication with the SerpApi API by submitting a request that includes the search engine as well as the search query that they wish to execute.

2. The request is received by the API service, which then verifies it by checking to see if the user submitting it has the appropriate rights and whether or not the format of the request is as expected.
3. The request is then sent from the API service to the relevant microservice to be processed, and this is determined by the type of query and the search engine being used.
4. The search query that is requested is carried out by the microservice by sending a request to the website of the relevant search engine, and utilizing automated web scraping techniques in order to retrieve the pertinent data from the page that displays the search results.
5. After the data has been extracted, the microservice will process it by removing any superfluous or extraneous details and transforming it into a structured data format.
6. Following this, the formatted data is sent from the microservice back to the API service. The API service then combines the data with any additional metadata and delivers it back to the user as a response to the original request made by the user.

Throughout the entirety of the process, SerpApi infrastructure automatically manages the scaling and allocation of resources to guarantee a high level of availability, reliability, and performance (SerpAPI: Google Search API, n.d.).

3.3 Big data Framework and Parallel Computing

Today's massive data volumes require big data frameworks and parallel computing. These frameworks enable rapid processing, analysis, and insight extraction from massive datasets. These frameworks let businesses overcome traditional computer limits by splitting jobs into smaller components and executing them simultaneously. Big data frameworks increase computational capacity, data processing speed, and real-time analytics by distributing workloads across several servers. They support data-intensive applications predictive analytics. Big data frameworks and parallel computing technologies are transforming data-driven decision-making and industries worldwide (Halas, 2017).

3.3.1 Parallel Data Processing System

The principles of divide and conquer are used in distributed computing, where different components of the parent task are completed on different machines and the results are combined. The term "parallel computing" refers to the use of a single CPU within a system or a collection of closely coupled systems through the use of effective and resourceful multi-threading processes. The process of distributed computing involves connecting a number of computer resources to a network and distributing computing workloads among them to increase computing power. When it comes to processing a significant volume of data in a short amount of time, distributed computing is significantly more valuable than traditional computing, since it is quicker and more effective (Buyya et al., 2016).

The processing capability of a standalone personal computer can also be increased to perform complex computations by adding additional processing units, which can execute a complex task by decomposing it into smaller tasks and performing each of those tasks simultaneously. They are frequently referred to as parallel systems. The computation speed increases with processor power. To learn how Apache Hadoop and Apache Spark are applied in big data analytics, it is helpful to have a basic understanding of parallel processing and distributed processing. There can be confusion between distributed processing and parallel processing, because in both cases computing is divided into smaller chunks. The memory architecture is what distinguishes distributed computing from parallel computing. Since distributed computing is disk-based rather than memory-based, parallel computing tasks can access the same memory space while distributed computing tasks cannot. Both one computer and additional computers are used to run different distributed computing jobs (Mittal et al., 2019).

Although there are some significant differences between Apache Spark and Apache Hadoop, both are free and open-source frameworks for parallel and distributed processing of large amounts of data. Apache Spark employs Resilient Distributed Datasets (RDDs), whereas Hadoop normally uses MapReduce to process data (Ahmed, 2022). Data files may be sorted across numerous computers by the distributed file system HDFS that is part of Hadoop. Contrarily, Spark does not offer a file system for distributed storage, but rather,

it mostly reads and processes data from other file systems. The performance of Spark and Hadoop might reveal their main differences. Hadoop excels at batch processing, but it struggles with iterative processing, which is why Spark was developed. The Spark program consistently runs around 100 times quicker in memory than Hadoop, and about 10x quicker on the disk.

The in-memory processing used by Spark is what makes it fast. Hadoop, in contrast, reads data from a disk during the subsequent iteration, making it substantially slower than Spark. To overcome the drawbacks of Hadoop, Spark was developed to be quicker and more effective (Chanda, 2021).

Spark is capable of supporting parallel operations and numerous sources, each of which comes equipped with a fault tolerance mechanism that can be cached. In addition to this, it is capable of representing a single data set that has numerous partitions. Spark is a distributed computing framework that can run a single or several concurrent interactive jobs across its master and worker nodes (Veiga et al., 2016). In Spark, a job can be broken down into one or more physical units for execution, and each of these jobs can be further subdivided into a more manageable set of tasks that are displayed on the stage. As a result, the job that is submitted can be carried out simultaneously. Spark divides the execution of jobs that have been submitted into two stages: the ShuffleMapStage and the ResultStages. The ShuffleMapStage is an intermediary stage in the DAG that serves the purpose of storing the output data in preparation for its use in subsequent stages as input data. This procedure comes to a close with the execution of the ResultStages, which assigns a function to one or more partitions of the target RDD (Ahmed, 2022).

To implement parallel algorithms, there are numerous parallel programming models available. These models serve as a roadmap for creating effective parallel code. There are different categories for parallel computing depending on the situation, including (Vázquez Benítez, 2015):

- ❖ Distributed computing is any computation involving numerous computers located far apart from one another, each of which plays a part in a computation issue.
- ❖ With cloud computing, access to resources for parallel computing is provide by companies like Amazon EC2, Microsoft Azure, and others.

Compute resources are made available on an hourly basis and can be expanded as necessary. For instance, Amazon offers EC2, EMR, and HPC.

3.3.2 Apache Spark

The UC Berkeley RAD Lab is responsible for the development of Spark (now called as AMP Lab). Matei Zaharia and his colleagues are the primary contributors. Its primary goal is to enhance Hadoop so that it could be used as a general-purpose framework (Buyya et al., 2016). It does this by extending MapReduce with more primitives and operators, which makes the algorithm more general (Salem, 2016).

Spark's ability to deliver low latency and high throughput for enormous data sets, in comparison to those of many other frameworks, is primarily responsible for the enormous amount of popularity it has garnered in the field of big data processing (Rahman et al., 2021). It is distinguished from other frameworks by a number of qualities that set it apart from the competition and make it desirable. Speed, powerful caching, deployment, real-time support, and polyglot support are some of the benefits of Spark. (Jaiswal et al., 2020).

Apache Spark is a free and open-source general-purpose cluster computing system that was developed with the goal of quickly processing large datasets. It offers high-level application programming interfaces (APIs) for the programming languages Scala, Java, Python, and R. Spark has evolved as an addition to Hadoop MapReduce, as a more user-friendly alternative, offering fast processing due to its complicated distributed methodology. This is in response to the increased need for parallel data analysis (Halas, 2017). In comparison to competing Big Data frameworks, such as Hadoop and storm, Spark possesses a number of advantageous characteristics. Spark is utilized by a wide variety of enterprises, including Yahoo, Baidu, and Tencent, amongst others. Resilient Distributed Datasets is one of the fundamental ideas behind Spark (RDDs). An RDD is essentially a collection of objects that cannot be changed and are distributed throughout a Spark cluster (Ramírez-Gallego et al., 2017).

3.3.2.1 Spark architecture

A Spark cluster utilizes a master/slave design and is comprised of the following three primary components (Inoubli et al., 2018) (Halas, 2017):

1. Spark Driver: This component is what a Spark cluster's slave node is called. It controls and monitors applications that are already running through an object known as Spark Context, which is maintained by it. It performs the functions as follows:

- ❖ A distinct process to carry out the functions of user applications
- ❖ Generates a Spark Context in order to arrange the execution of jobs and to negotiate with the cluster manager

2. The Cluster Manager: is the component that is accountable for coordinating the workflow of the application that is given to the workers by the Driver Program. Additionally, it controls and supervises all of the cluster's resources and reports back to the Driver Program regarding the state of those resources. Some type of Cluster Manager:

- ❖ YARN
- ❖ Spark Standalone
- ❖ Mesos

3. Worker Nodes: Throughout the process of putting a Spark program into action, each Worker Node acts as a container for a single operation. It performs the functions as follows:

- ❖ Carry out activities that have been scheduled by the driver
- ❖ Store results of computations in memory, or on disk
- ❖ Communicate with the various storage systems

Figure 3.1 below illustrates Apache Spark architecture

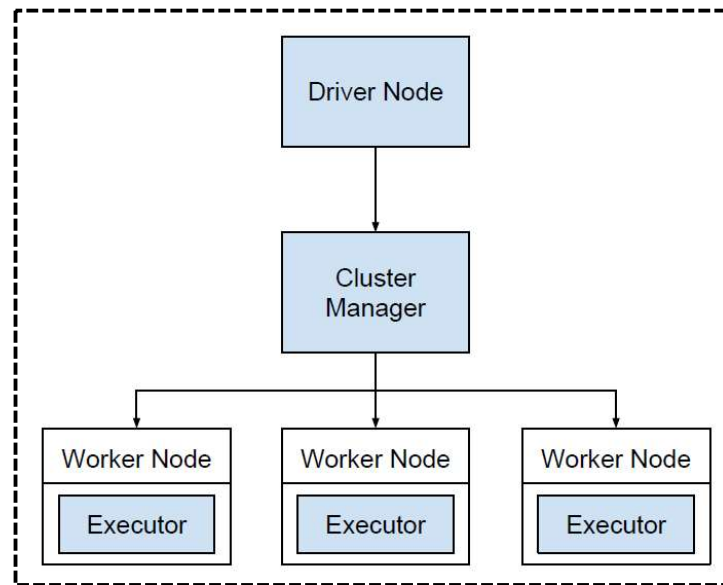


Figure 3.1. Apache Spark architecture (Inoubli et al., 2018)

Launching Spark apps can take place either locally on a single machine or remotely within a cluster where they will share the resources of the cluster. In local mode, the driver, executors, and tasks all operate on the same Java Virtual Machine (JVM) within the same Java process. While running in cluster mode, Spark application needs to connect to a cluster manager that is responsible for allocating resources among the many apps.

Apache Spark Driver includes the following additional components, each of which is accountable for the translation of user code into actual jobs that run on cluster (Buyya et al., 2016) (Feng, 2018):

- ❖ **Spark Context:** represents the connection to a Spark cluster, and it may be used to build RDDs, accumulators, and broadcast variables on that cluster. Moreover, it can be used to retrieve information about a Spark cluster.
- ❖ **DAG Scheduler** is responsible for computing a DAG of stages for each job and submitting them to Task Scheduler. It then selects optimal locations for tasks (based on cache status or shuffle file locations), and calculates the minimum schedule to run the jobs.
- ❖ **The Task Scheduler** is in charge of submitting tasks to the cluster, running them, and retrying them in the event that they fail.

3.3.2.2 Apache Spark Components

Apache Spark is made up of several primary parts. The most important ones are Spark core and upper-level libraries. Spark core is capable of accessing data stored in any Hadoop data source and can run on a variety of cluster managers. In addition, a large number of packages that are compatible with Spark core and upper-level libraries have been developed. The various aspects of Spark are broken down and illustrated in Figure 3.2., and the components that build the Spark ecosystem are listed below (Halas, 2017) (Buyya et al., 2016) (Salloum et al., 2016):

❖ Apache Spark Core

Spark Core is the fundamental general execution engine for spark platform, and it is the foundation upon which all of the other functionality is constructed. It offers computing in memory as well as the ability to reference datasets located in external storage systems.

❖ Spark SQL

Is a component built on top of Spark Core that introduces a new data abstraction known as SchemaRDD. This abstraction offers support for structured, unstructured, and semi-structured data types.

❖ Spark Streaming

In order to carry out streaming analytics, Spark Streaming makes use of the quick scheduling power offered by Spark Core. It does RDD (Resilient Distributed Datasets) transformations on the mini-batches of data that it ingests and conducts those transformations on those mini-batches of data.

❖ MLlib stands for "Machine Learning Library"

It stands for "Machine Learning Library", and it is a distributed machine learning framework that is built on top of Spark. This is because Spark's design is based on distributed memory. According to the benchmarks, it is something that was done by the creators of MLlib in competition with the Alternating Least Squares (ALS) implementations. Spark MLlib is nine times faster than the disk-based version of Apache Mahout that is supported by Hadoop (before Mahout gained a Spark interface).

❖ GraphX

GraphX is a framework for the distributed processing of graphs that is built on top of Spark. It offers an API for expressing graph computation that, through making use of the Pregel abstraction API, may model user-defined graphs. In addition to that, it offers a runtime that is optimized for this abstraction.

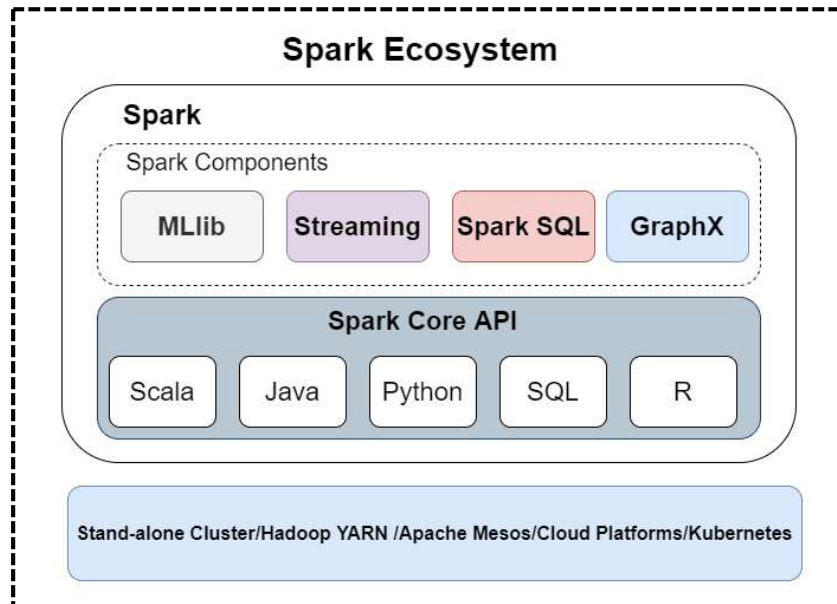


Figure 3.2 Components of Apache Spark Ecosystem (Halas, 2017)

3.3.2.3 Apache Spark Cluster Management

As volume and complexity of data continue to grow, distributed computing systems like Spark have become increasingly popular for processing large-scale data sets. However, running Spark applications across a cluster of machines requires effective resource management and job scheduling, which is why Spark Cluster Managers play a vital role in enabling efficient and reliable processing of big data. The available spark cluster managers are (Mittal et al., 2019) (Hamstra & Zaharia, 2013):

1. **YARN:** YARN (Yet Another Resource Negotiator) is a popular cluster manager for managing resources in a Hadoop cluster, and it is also supported in EMR for managing Spark jobs. YARN allows multiple applications to share the same cluster resources, making it a good choice for running Spark jobs alongside other Hadoop ecosystem applications.

2. **Standalone:** The Spark Standalone Cluster Manager is a lightweight cluster manager that is bundled with Spark. Provides basic resource management and scheduling capabilities.
3. **Kubernetes:** EMR also supports Kubernetes, a popular container orchestration system that can be used as a Spark Cluster Manager. Kubernetes provides advanced container scheduling and orchestration capabilities, making it a good choice for running Spark jobs in a containerized environment.

Via the use of the spark-submit script, Spark offers a single method of running programs on its platform. The script can either submit the application in local mode or connect to any of the supported cluster managers, and its behavior is determined by the inputs that are supplied to it. The driver program is initiated by the spark submit script, which then makes a request to the cluster manager for resources in order to activate executors. Throughout the process of an application being executed, the driver is responsible for delegating tasks to executors (Hamstra & Zaharia, 2013).

3.3.3 MongoDB Database

MongoDB is a document-oriented, NoSQL database system that is employed for the purpose of managing large amounts of data. The fact that it is a NoSQL database indicates that its structure for the storage and retrieval of data is distinct from that of conventional SQL databases, which rely on tabular relations. The documents that make up MongoDB's collection system can range in size and content matter, depending on the document (Osipov, 2019). MongoDB utilizes java Script Object Notation (JSON) file format for data storage. JSON is capable of handling all of the fundamental data kinds, including arrays, strings, numbers, and Boolean values, among others (Edward & Sabharwal, 2015).

The data is saved in pairs of fields and values, similar to how objects in JavaScript are stored. The id field of a MongoDB document is what distinguishes it from every other document in the database. If there are no id fields defined, then the system will generate them automatically. In MongoDB, a collection is the collective noun for a group of documents. Documents and collections can be thought of as being analogous to rows and tables, respectively, in relational databases (Le, 2022). According to its official

website, it is "The most popular database for modern apps". Scalability, replication, quick performance, distributed I/O, and non-blocking IO are among of MongoDB's many advantages. MongoDB is utilized to store, in situations where an amount of data is too large (Kholod, 2021). Bigdata, Content Management and Delivery, Mobile and Social Infrastructure, User Data Management, and Data Hub are some of the applications that make use of MongoDB (Chauhan, 2019).

Due to the fact that MongoDB does not utilize a schema, the structure of documents can differ from one collection to another. When referring to a document model, the term "schemaless" indicates that the model does not need to be pre-structured in any way. In the event that a new field is necessary at a later time, it is possible to construct it as needed without having to completely rebuild the structure of the document, and it will not have any impact on any other documents. In order to boost its throughput in a horizontally scalable environment, MongoDB uses a technique known as sharding, which involves distributing large amounts of data across numerous servers. sharding works to evenly distribute the burden across all of the servers so that data processing and maintenance downtime are not disrupted. In addition, MongoDB makes use of an in-memory database that stores data in the host's random-access memory (RAM) in order to improve performance. When compared to SSDs and disks, having data stored in RAM allows for far quicker access to the data. Although storing data in RAM is associated with a significant risk of losing data in the event of a power loss, MongoDB eliminates this risk by generating a replica set to on-disk storage while simultaneously preserving processing performance. The process by which MongoDB replicates data both in memory and on disk is illustrated in Figure 3.3. (Thapa, 2022) (Edward & Sabharwal, 2015).

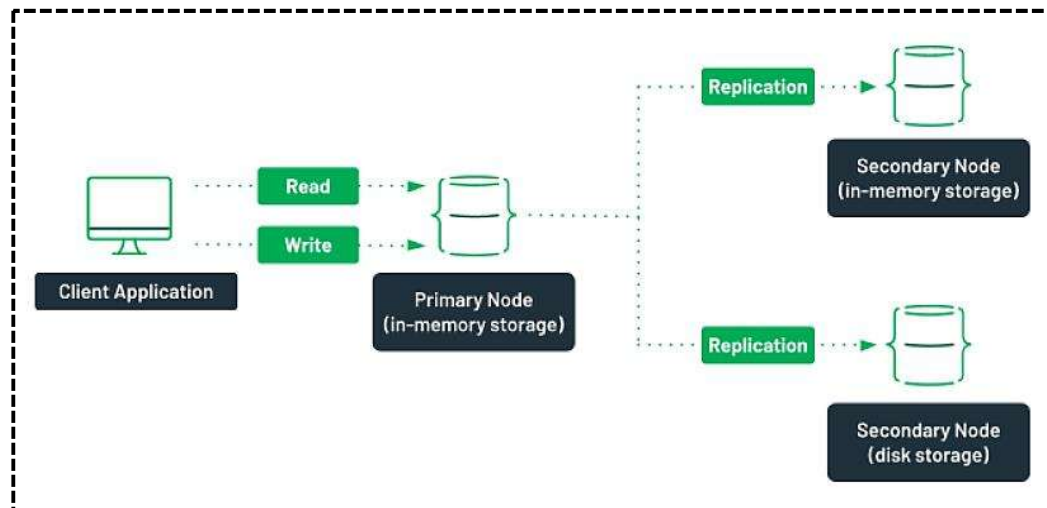


Figure 3.3 Replication of Data in Memory and Disk in MongoDB (Thapa, 2022)

3.3.3.1 Architecture of MongoDB Deployment

When storing all data files in MongoDB on distributed setups spanning multiple machines, ways to partition data files among multiple systems must be investigated. The internal organization of MongoDB is shown in Figure 3.4. Sharded MongoDB comprises three components (servers) as follow: (Shin et al., 2022) (Abdel-Fattah et al., 2022).

- ❖ **Config server:** The Config server maintains and saves the metadata, which contains information about which shards of data are stored on which shard servers.
- ❖ **Router server (mongos):** The Router server uses the metadata stored in the Config server to route client requests to the relevant shard server. Data shards are stored on each shard server, separated from full data sets, and replicated across the cluster at a speed that is set to three times the previous replication factor by default.
- ❖ **Shard server:** Is the most important aspect influencing the performance of data sharding in MongoDB.

For the purposes of concurrent access and efficient read/write operations, the data in MongoDB is spread and stored on more than one server, which is referred to as a shard server. The process in question is known as "Sharding." It is implemented by separating the MongoDB server into mongos, which is a

series of routing servers, and mongod, a set of data servers, which are responsible for routing operations (Sangat et al., 2018).

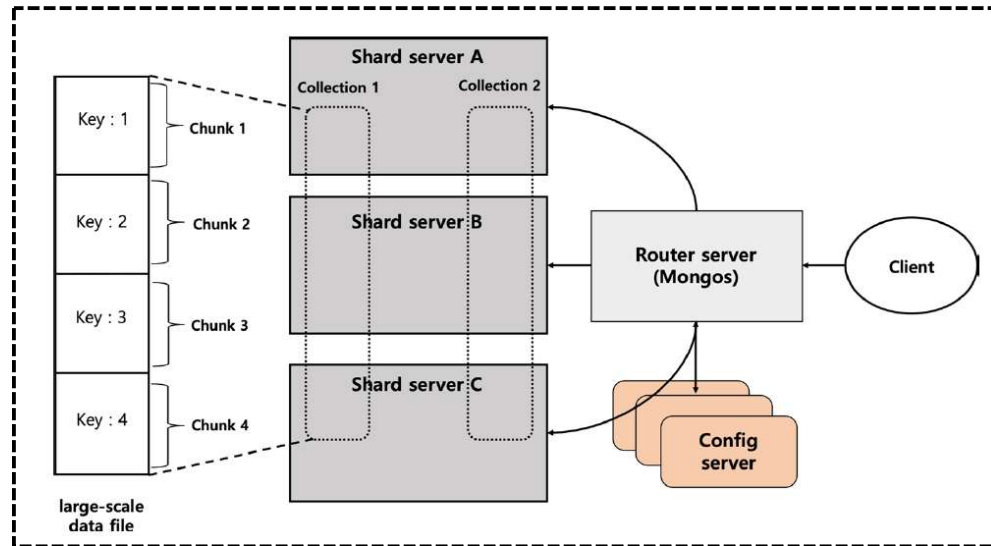


Figure 3.4 Architecture of Sharded MongoDB (Shin et al., 2022)

In a replica set deployment, the architecture is typically consisting of the following components:

- ❖ **Replica Set Members:** These are individual MongoDB instances (nodes) that host the data. Each member can act as either a primary or a secondary member. The primary member accepts write operations from clients, while secondary members replicate data from the primary and can serve read operations. The primary member is responsible for managing failover and electing a new primary if the current primary fails.
- ❖ **Replica Set Configuration:** Configuration information, such as the list of replica set members and their roles, is stored in a configuration document called the replica set configuration. Each member of the replica set maintains a copy of the configuration document, and it is used for coordination and communication among members.
- ❖ **Client Application:** Client applications connect to the replica set using connection strings that specify the addresses of the replica set members.

3.3.3.2 Clustering in MongoDB Atlas

When a Cluster instance is created in MongoDB Atlas, a group of servers is created that work together to store and manage data. These servers are

distributed across multiple geographic regions to ensure high availability and low latency for application. Each Cluster instance consists of a primary node and one or more secondary nodes. The primary node is responsible for handling all write operations, while the secondary nodes are responsible for replicating data from the primary node and handling read operations. If the primary node fails, one of the secondary nodes is automatically promoted to become the new primary node. This ensures that application can continue to read and write data even in the event of a server failure as shown in Figure 3.5 (Hows et al., 2014).

In addition to high availability, security features, such as, encryption at rest and in transit, network isolation, and authentication are provided by MongoDB Atlas to ensure that data is protected from unauthorized access. A number of tools and services are offered by MongoDB Atlas to help manage and monitor the Cluster instance. The performance and health of the database can be monitored using MongoDB Compass (Bernard & Kintziger, 2021).

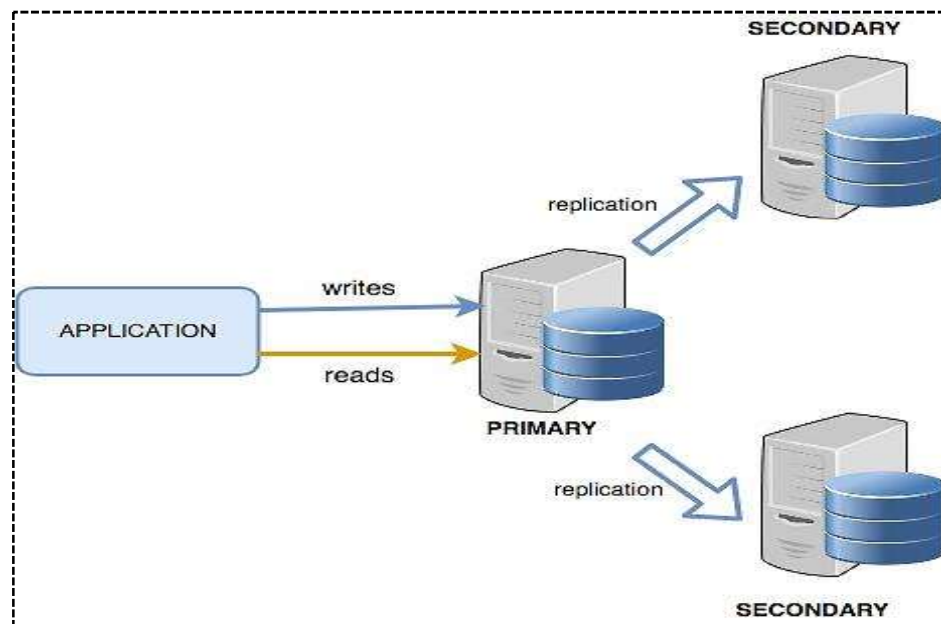


Figure 3.5 Deploy a MongoDB Cluster (Hows et al., 2014)

In MongoDB Atlas, clusters provide several benefits over standalone MongoDB instances, including (Edward & Sabharwal, 2015):

- **Scalability:** Clusters allow scaling of database horizontally by adding more instances to handle increased workload.

- Availability: Clusters provide high availability by replicating data across multiple nodes, so that if one node goes down, application can continue to access the data from another node.
- Fault tolerance: Clusters provide fault tolerance by ensuring that data is distributed and replicated across multiple nodes, reducing the risk of data loss due to hardware failures or other issues.
- Security: Clusters in MongoDB Atlas provide advanced security features such as network encryption, user authentication, and role-based access control, helping to protect data from unauthorized access.
- In MongoDB Atlas, different types of clusters can be created, such as replica sets and sharded clusters. Replica sets deployment are clusters of MongoDB instances that provide high availability and fault tolerance by replicating data across multiple nodes. Sharded clusters deployment are clusters of MongoDB instances that allow partitioning of data across multiple nodes, enabling horizontal scaling of database (Edward & Sabharwal, 2015) (Osipov, 2019).

3.3.4 Integrating Spark with MongoDB

Capturing, storing, and analyzing dynamic unstructured and semi-structured data that comes from the web is becoming an increasingly pressing necessity. Due to the comparable expansion of semi structured data on the current web, NoSQL data stores have been developed to improve scalability, performance, and availability, Parallel analysis has been made easier with the development of large-scale data processing tools (Sangat et al., 2018).

MongoDBConnector for Spark (SDBC) is authorized version "V1.0" of the MongoDBConnector for Apache Spark was made available for download on June 28th, 2016. This connector ensures that MongoDB and Apache Spark work together in a coordinated and seamless manner. It does this by successfully utilizing the aggregation pipelines and secondary indexes that are available in MongoDB. In addition, it limits the amount of data that is transferred throughout the cluster by linking RDDs to the source MongoDB node. This allows it to maximize performance across huge dispersed data sets (Sangat et al., 2018).

According to the findings of a comparative study, MongoDB is considered more effective than PostgreSQL in handling spatial data. A MongoDB

connector for Spark is offered as a means of providing support for MongoDB on Spark. Data may be moved from Spark, serving as the data processing engine, to MongoDB, serving as the data storage engine, using this connector. Furthermore, data can be written to MongoDB from Spark. The connection also enables the use of aggregation pipelines, which filter data and perform aggregations in MongoDB instead of loading the entire dataset in Spark. This reduces the amount of data that needs to be moved from MongoDB to Spark. Figure 3.6 illustrates how distributed MongoDB is utilized as a Spark storage engine (Shin et al., 2022) (Alberto, 2021).

The Spark MongoDB connector is responsible for the creation of numerous concurrent connection pools within the MongoDB database. In this configuration, each incoming connection is assigned one thread, and each parallel-processing thread is executed within its own core (Abdel-Fattah et al., 2022). The connector makes it possible for Spark to perform distributed data processing on MongoDB. Scala, Java, Python (through PySpark), and R are some of the different programming languages that are supported. It grants Spark context the ability to access key spaces that already exist in a MongoDB database (Shah, 2019).

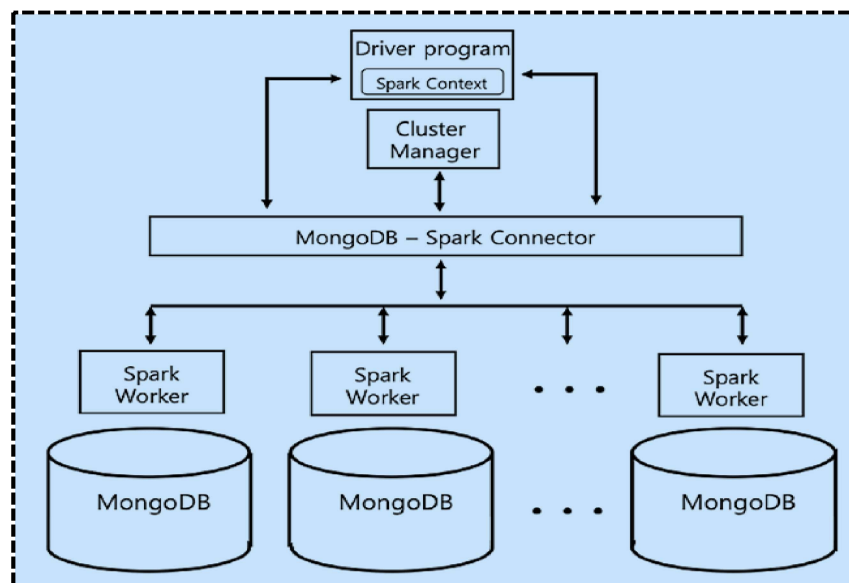


Figure 3.6 Architecture of MongoDB Spark Connector (SDBC) (Shin et al., 2022)

The middleware MongoDB Connector for Spark is used to realize the seamless connection between the computing engine and the database. This is

done so that the respective benefits of the computing engine and the database can be utilized to their maximum potential. The Spark computing engine is provided by MongoDB rather than HDFS, and the benefits that it offers in terms of aggregation and auxiliary indexing are exploited to their full potential. When performing data analysis, Spark first extracts and filters data through MongoDB. Then it only reads the data that is required by Spark operation. This increases memory consumption efficiency while reducing data redundancy and boosts the processing capability of Spark memory (Zhang et al., 2020).

3.4 Cloud Service and Distributed Systems

Cloud computing has emerged as a significant invention in the field of Information and Communication Technology (ICT), and has the potential to transform the way in which information resources are utilized and distributed. This innovation is seen as a fresh approach that can provide a new information infrastructure, which has the potential to significantly contribute to the expansion of the economy in the years to come. When it is uploaded to a cloud platform, a web application becomes part of a distributed system that is made up of numerous service layers. The cloud infrastructure service layer is the lowest service layer that a developer may work on in cloud computing. Since a cloud platform is a distributed system, it is quite likely that any problems with the underlying infrastructure will also affect any web applications that are deployed to such platforms. This is due to the distributed nature of cloud platforms (Lula et al., 2021).

Cloud Computing services are now offered by the majority of the world's largest software businesses, including Google, Microsoft, and Oracle, amongst others. In addition, cloud computing has fundamentally altered the traditional model for the delivery of services. This has made it possible to provide virtualized services that can be delivered via the internet and scaled in terms of their processing power and storage capacity. The processing of big data can also be supported by cloud computing, which offers powerful storage, compute, and distributed capabilities. It is necessary to implement new data analysis algorithms as well as new methodologies in order to successfully manage the exponential increase in data volume and the requirements of massive-scale analytics in order to realize the full potential of big data. Due to this, the economies of scale are one of the fundamental benefits that comes along with providing services over the cloud. When a service provider makes use of the

cloud's underlying infrastructure, it is possible to provide services that are superior in quality while also being more affordable and dependable (Hill et al., 2015).

The term "cloud services" refers to any hardware, software, or platforms that are hosted by third-party vendors and made accessible to customers through the use of the internet. The utilization of cloud computing makes it possible to provide a variety of services via the internet. Companies from all over the world are compete for the top spot in the cloud computing industry's information technology (IT) area. Microsoft Azure, Google Cloud, AppScale, Heroku, and Amazon Web Services (AWS) are the most widely used PaaS platforms. AWS (Amazon Web Services) is a well-known cloud service that allows internet applications to be launched and expanded. Clients have the ability to exercise full control over the AWS resources that are ordered to execute the apps that they have uploaded, while at the same time having large-scale installs managed automatically (Ruparelia, 2016) (Kumar & Kumar, 2019).

3.4.1 AWS Cloud Service

The three levels that make up the AWS architectural stack are referred to as global infrastructure, foundation services, and platform services, respectively. Figure 3.7 provides a visual representation of these three tiers, demonstrating that relational databases, data warehousing analytics, various forms of applications such as email and streaming, management tools, and

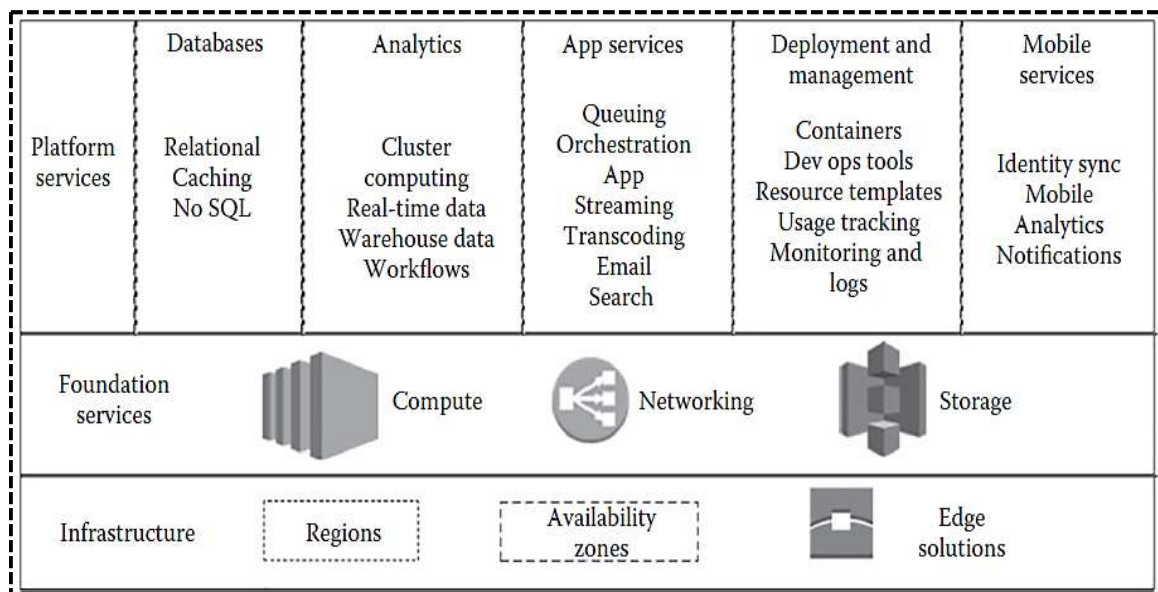


Figure 3.7 Architectural of AWS (Srivastava & Khan, 2018)

mobile services are all included in the category of platform services. Computing, storage, and networking are all included in the core services, which are more general in nature. Infrastructure is the implementation that is based on the internet that provides these services (Fox & Hao, 2017) (Ruparelia, 2016) (Srivastava & Khan, 2018).

3.4.1.1 Platform as a Service (PaaS)

PaaS is a cloud computing service model that provides a platform for building, deploying, and managing applications. PaaS offerings often include tools and services for building, testing, and deploying applications, as well as runtime environments for executing those applications (Fox & Hao, 2017). MongoDB Atlas is a fully managed cloud-based database service that provides automatic scaling, backups, monitoring, and many other features that ease the use of MongoDB in the cloud. MongoDB Atlas allows to deploy, scale, and manage MongoDB clusters without having to worry about the underlying infrastructure. It is primarily used for managing the operational aspects of MongoDB database, such as performance tuning, backup and restore, and security configuration. MongoDB Atlas is considered a fully managed database-as-a-service (DBaaS) offering, which falls under the umbrella of platform-as-a-service (PaaS). DBaaS is a cloud computing service model, where the database is managed and maintained by a third-party service provider, such as, MongoDB Inc. in the case of MongoDB Atlas (Hill et al., 2015).

3.4.1.2 Infrastructure as a service (IaaS)

Amazon Elastic Compute Cloud (EC2) offers scalable computing power in AWS Cloud and eliminates the upfront hardware investment requirement, enabling applications to be built and deployed more quickly. Using EC2, virtual servers may be launched as needed, and security, networking, and storage can all be set up and controlled. Features of EC2 include various CPU, memory, storage, and networking capacity configurations known as instance (Hill et al., 2015).

3.4.1.3 Software as a Service (SaaS)

An Amazon Machine Image (AMI) is a prepackaged template that makes the software start an instance in EC2, including operating system and extra applications. It provides the framework for building cloud-based virtual servers.

AMIs can be modified or distributed across users and are intended to give instances a consistent starting point. The root file system, launch permissions, and block device mapping are all contained within them. Users can quickly reproduce and deploy instances with particular configurations and pre-installed software by using AMIs, which reduces setup time and effort (Mittal et al., 2019).

3.5 Virtualization Concepts

Virtualization is commonly believed to be the creation of an abstraction layer over physical hardware known as Virtual Machine (VM). Simply, a virtual machine is a virtual representation of a physical computer with its own operating system that cannot directly interact with the underlying hardware. It needs a layer known as a hypervisor. Hypervisor is responsible for allocation of resources like storage and computing power to the virtual machines. The principal benefits of implementing virtualization are evident in the decreased expenses associated with electricity, IT equipment, and storage capacity. In addition to offering strong security and resilience to faults (Ganesh et al., 2022).

3.5.1 Virtualization Tools (VirtualBox)

In hardware-based virtualization, the two widely used software are VirtualBox and Vagrant. While they both essentially provide the same functionality, they have a considerably different way of working. VirtualBox, however, is a powerful virtualization product which is rich in features and performance. It mainly runs an operating system on a single machine and allows free switching between the OS instances running. Isolated developers prefer VirtualBox setup which requires more effort in setting up but is fairly easy to use and manage. VirtualBox is a tool more popular due to ease of use and multiple features it offers by Oracle Corporation, is an advanced, free and open-source hosted hypervisor which is used for x86 virtualization. It is a hardware assisted virtualization tool which enables users to load multiple OS under a singular hosting operating system. Each hosted VM is configurable and can also be put into networks which provides connection between the hosted OS and the hosting OS (Agarwal et al., 2020).

For machines with AMD or Intel CPUs, Oracle's VirtualBox is an extremely potent application for virtualizing 32- and 64-bit operating systems. The operating systems Windows, Linux, and Mac can all run this software

(Đorđević et al., 2022). The following are the technical prerequisites needed to run this software:

- ❖ 32- or 64-bit operating system paired with an AMD or Intel hardware.
- ❖ 512 MB or more RAM is required, contingent on the quantity and kind of virtualized operating systems.
- ❖ Since the program only permits the use of half of the RAM on the base system, the RAM memory space allotted to the virtual machine environment may reach half of its maximum capacity.
- ❖ The virtual machine environment's accessible hard drive space should have a minimum size of 8GB.

Data sharing across virtual machines is quick and simple with VirtualBox as shown in Figure 3.8. VirtualBox is a software that is open source, which means that it may be updated with new features and typically solves issues rapidly (Sivaraj et al., 2021).

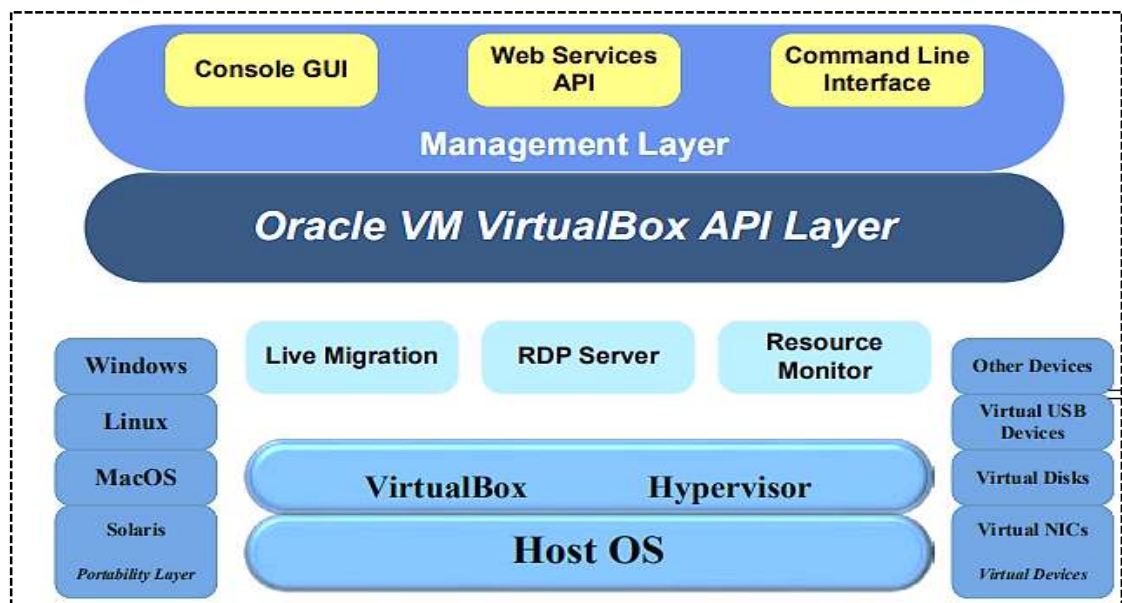


Figure 3.8 Architectural of VirtualBox (Sivaraj et al., 2021)

3.5.2 Virtualization in Cloud Computing (EC2)

Virtualization is the foundation for cloud computing's Infrastructure as a Service (IaaS) delivery model, which isolates workstations, data, networks, and apps from hardware limitations. Since cloud computing offers cost-effective, adaptable, and agile services at the software, platform, and infrastructure levels,

it is gaining popularity among IT firms. Within the infrastructure as a service (IaaS) framework, cloud service providers deliver network, compute, storage, and database services through the internet. All cloud services are constructed on top of IaaS, with PaaS and SaaS following suit. IaaS is primarily characterized by virtualization and elasticity (Singh, 2018).

Through the utilization of Amazon Elastic Compute Cloud (EC2), users can effectively rent computational resources as needed. Using the Xen hypervisor, which enables Amazon EC2 to generate Virtual Private Servers on demand. Operating systems that have been tuned for paravirtualization are used by these virtual servers. The virtual machine can be saved as an Amazon Machine Image, or AMI, once the user has made all the necessary customizations. AMI is a system image designed specifically for use with EC2. These AMIs can be customized by users using their own software programs, making an AMI suitable for any task (Saini & Behl, 2020),(Choudhary et al., 2021). To make the most of the processing and storage, the computer is virtualized and placed in a clustered computing environment under the supervision of cloud computing. Through rigorous monitoring of resource workload and available physical resources, as well as continuous managed solution provision, the cloud will offer dynamic resource allocation to ensure effective system usage (Singh, 2018).

Upon EC2 adoption, Amazon provides a service that is built on the "cloud computing model." Under this "pay as you go" model, consumers can use the network infrastructure to borrow a portion of the desired size. Amazon EC2 server instances may be utilized whenever needed, and for as long as required (Chythanya et al., 2019). Everything in the original EC2 architecture, including network, storage, management, security, and monitoring, is centralized on the server, as shown in Figure 3.9. This means that server resources are being used (Saini & Behl, 2020).

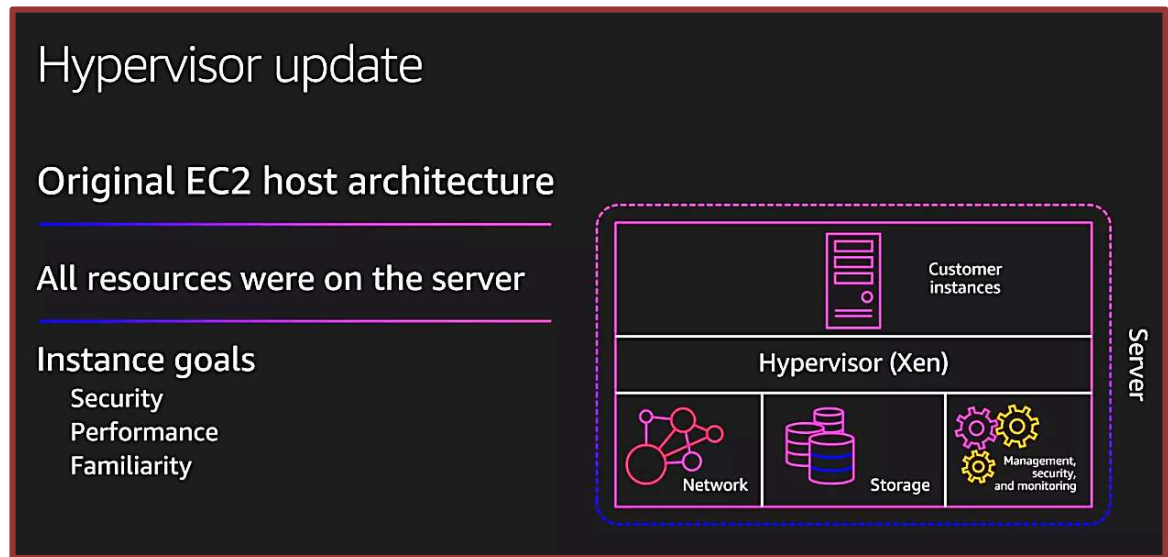


Figure 3.9 EC2 Architecture (Saini & Behl, 2020)

3.5.2.1 Amazon EC2 Main Feature

The main features offered by EC2 service are stated below (Choudhary et al., 2021).

- 1) One kind of virtual computing environment is an instance. A platform called EC2 lets set up server on the cloud. Servers are typically referred to as instances in AWS. An instance can be accessed at any time and from practically anywhere because it is virtualized.
- 2) A variety of instances with varying CPU, memory, and network specifications are available through EC2. This lets the user choose the optimal instance according to CPU, RAM, and network configuration.
- 3) An Amazon Machine Image (AMI), which is a pre-configured template of operating systems and apps, is used to build an instance. This reduces the overall instance distribution time as well as the need to configure every step of the application post-OS deployment.
- 4) The AWS cloud is located across numerous nations and continents. one can use availability zones and regions to determine where to put instance.
- 5) With regard to the security of the instance, the private key is received from AWS, while the public key is retained by them. Access to the instance is not granted until the secret key is provided.

6) Upon the creation of an EC2 instance, it is assigned one private IP address and one optional public IP address by AWS. In contrast, the public IP address is dynamic and may change if the instance is shut down or restarted. Obtaining a static public IP address is possible with elastic IP addresses.

3.6 Complex Network

Complex network can be characterized as a graph having a large number of features that do not present in conventional networks like network or random graphs, but primarily appear in graph models of real systems (Silva & Zhao, 2016). This definition comes from the theoretical principle of networks, which states that a complex network can be thought of as a graph. The investigation of complex networks was initiated at the turn of the 21st century and has since developed into a dynamic subfield of the scientific study community. The experimental study of real-world networks, including technical networks, biological networks, computer networks, climatic networks, road networks, social networks, and brain networks, is a major source of inspiration for it (Burguillo, 2018). In general, it is heavily inspired by the study of real-world networks. The following are some examples of measures that can be used for complex networks:

3.6.1 Complex Networks Measurements (Network Level Metric)

Below is an explanation of some measurements of complex networks at network level:

- ❖ **Average Clustering Coefficient:** often known as the ACC, is a metric that determines the extent to which nodes in a complex network have a tendency to cluster or form groups that are densely associated. It is calculated as the mean of the local clustering coefficients of each of the nodes that make up the network. The local clustering coefficient of a node determines the degree to which its neighbors are related to one another and other nodes in the network. To be more specific, it is the ratio of the number of connections between a node's neighbors to the maximum number of connections that are conceivable between these neighbors. A node that has a low local clustering coefficient has neighbors that are relatively separated from one another, in contrast to a node that has a high local clustering coefficient with neighbors that are extensively associated with one another (Johansson, 2020).

- ❖ **Average Path Length (L):** The average path length is the average number of steps required to travel from one node to another in a network. It is a measure of the efficiency of a network. A short average path length indicates a more efficient network with fewer steps required to travel between nodes. Average path length can be used to compare the efficiency of different networks. It is determined by adding up the length of the shortest path between every pair of nodes, then dividing that total by the total number of paths between pairs. The value of this metric indicates how well-connected the network is (Sporns, 2022).
- ❖ **Network Diameter:** The network diameter is the longest shortest path between any two nodes in a network. It is a measure of the distance between nodes in the network. A small network diameter may indicate a well-connected network with short paths between nodes, while a large network diameter may indicate a less connected network with longer paths between nodes. It indicates the maximum number of hops that are required to get from one node to another in the network. It is a measurement of how large the network is in its whole (Saxena & Iyengar, 2020).
- ❖ **Density (D):** The density of a network is a measure that indicates how interconnected the nodes are within the network. A denser network has a higher density value, indicating a greater number of connections among its nodes. It is stated in the following equation, which indicates the total number of feasible linkages in that network divided by the total number of network edges (Kartun-Giles & Bianconi, 2019).

$$D_G = \frac{2(E(G))}{N(N-1)} \quad \text{..... (3.1)}$$

Where: $E(G)$ is the total number of edges, N is the total number of nodes, and $(N-1)$ is the total possible connections between nodes.

3.6.2 Complex Networks Measurements (Nodes Level Metric)

It is possible to analyze the performance of the nodes that make up a network, which includes the following factors:

- ❖ Degree Centrality (DC) of a node: It is a measure of how many links are established between individual nodes in a network. In other words, the total number of edges that are connected to a specific node (also known as the degree of a node). In terms of network visualization, the size of the vertices is determined by the degree (a greater degree corresponds to a larger size) (Saxena & Iyengar, 2020).
- ❖ Betweenness Centrality (CB): Betweenness centrality is a measure of the importance or centrality of a node in a network based on its position in facilitating communication or interaction between other nodes. It quantifies the extent to which a node lies on the shortest paths between pairs of other nodes in the network. Nodes with high betweenness centrality can be seen as critical for maintaining efficient communication or information flow within the network. They act as bridges or intermediaries between different parts of the network. The following is one possible way to build the CB of node j: (Saxena & Iyengar, 2020).

$$C_B(j) = \sum_{i \neq j \neq k} \frac{\sigma_{ik}(j)}{\sigma_{ik}} \quad \dots\dots\dots (3.2)$$

Where:

$C_B(j)$ is the betweenness centrality of node j.

σ_{ik} is the total number of shortest paths between node i and node k.

$\sigma_{ik}(j)$ is the number of those shortest paths that pass-through node j.

The summation is taken over all pairs of nodes i and k (excluding j).

- ❖ Closeness Centrality (C_C) refers to the process of adding up the number of different network nodes that are directly connected to a certain node, which can be illustrated as following (Yang et al., 2020):

$$C_C(i) = \frac{N-1}{\sum_j d_{ij}} \quad \dots\dots\dots (3.3)$$

Where d_{ij} represents the distance between the i and j. This is a measurement of the significance of a node in a network that is based on its proximity to all of the other nodes in the network. A node is said to have a high proximity centrality if it is one of the few in the network that

can connect to every other node in the network with a limited number of hops. This type of node is vital to the network's ability to function well as a whole. This metric is essential for determining which nodes may be crucial in terms of the flow of information or resources, as well as which nodes may be more influential as a result of their capacity to rapidly disseminate information. It is determined by taking the inverse of the average distance that each node in the network has to travel to reach every other node in the network.

- ❖ Clustering Coefficient (C_O): is a measurement that reveals the degree to which authors tend to work in groups and collaborate. The formulae that can be used to calculate each author's C_O value are as follows (Yang et al., 2020):

$$Co_{(i)} = \frac{2|\{l_{jk}: n_j, n_k \in N, l_{ik} \in E\}|}{ki(ki - 1)} \dots\dots\dots (3.4)$$

Where:

E is the number of edges between the neighboring nodes of the author, and K is the total number of neighboring nodes connected to the author.

- ❖ Eigenvector Centrality: Nodes that have neighbors that have a higher degree of Eigenvector Centrality have a greater degree of power than nodes that have neighbors that have a lower degree of Eigenvector Centrality. The Eigenvector Centrality measures a node's contribution to the overall degree centrality of its neighbors (Saxena & Iyengar, 2020).

3.6.3 Collaboration Networks

"Collaboration networks" are a type of complex network made up of actors with similar interests, as sportsmen in the same sport or actors in a generic film. This also applies to teams of scientists known as coauthors who worked together to generate one or more scientific publications (SP) (Scherbakova & Bredikhin, 2021).

3.6.4 Co-authorship network

The examination of co-authorship networks illuminates the extent of idea, knowledge, as well as technology entanglement, and defines the composition of the scientific community. Different degrees of collaboration might be taken into consideration in this situation, including those between nations, within nations, within organizations, between scientific fields, within a broad subject, inside journals or conference proceedings, etc. Different perspectives on the SP-set are represented by co-authorship networks and citation networks. Generally speaking, bibliographic databases (DB) are the source of information regarding SPs. In co-authorship networks, nodes are SP-authors, and links are established by their co-authorship of one or more SPs. In citation networks, SPs are the nodes, and oriented links between nodes are established according to a binary citation relation. Collaboration and citation conventions do, however, have an impact on one another (Savić et al., 2019).

3.7 Visualization

It is possible to describe data visualization as the process of displaying data using graphical representations such as graphs, charts, and pictorial forms. Data, trends, and patterns in the data can be made sense of through a technology called data visualization. It is possible for decision-makers to see analytics in a visual form by data visualization, which also makes it possible for decision-making and the determination of courses of action depending on what is seen in the visualization.

3.7.1 Cytoscape Software

Cytoscape is a piece of open-source software that was developed in Java and built on the platform of NetBeans for the purpose of graph visualization and modification. It was designed to help users visualize complex networks in a multi-layered fashion. An existing network can be converted into a multi-layered network by using the plugin, which provides access to a range of configuration options. The plugin is compatible with a number of different layout algorithms and gives the user the ability to select which aspect of the network should serve as the layer (Sadiku et al., 2016). When it comes to the visualization of general large-scale networks, Cytoscape is a more qualified candidate than its competitors due to the fact that it is a free open-source network visualization and exploration tool, it makes better use of memory and

is easier to use. Users are able to investigate the connections that exist between the nodes (vertices) and edges (links) in a network using its intuitive graphical user interface (GUI), which is provided for the purposes of developing, analyzing, and visualizing networks (Lima et al., 2018).

Users are able to conduct a wide range of different forms of network analysis with the help of Cytoscape. By using Cytoscape, many different network analysis tools are provided, which are tools for measuring network properties such as centrality, clustering, degree distribution, and community structure. In addition, it provides powerful visualization tools that enable users to create interactive and customizable visualizations of network data. The software enables the user to select from a number of different visualization options, such as circular layouts, node-link diagrams, and matrix visualizations. Cytoscape has numerous layout algorithms that allow users to arrange the nodes and edges of a network in different ways. ForceAtlas, Fruchterman-Reingold, Yifan Hu, and OpenOrd are some of the algorithms that fall under this category (Kartun-Giles & Bianconi, 2019).

CHAPTER FOUR

ACDN System Design Methodology

4.1 Preface

This chapter presents the design of the Academic Capital Distribution Network (ACDN) proposed system and the research method followed. The parallel processing techniques alongside the cloud computing techniques used in this thesis are also described. Moreover, details of the data collection process, Storage, processed, and network creation and analysis strategies are illustrated are dynamically executed, ensuring adaptive resource allocation based on workload demands.

4.2 System Architecture

This section is dedicated to the comprehensive approach employed within this thesis to illustrate the complex scholarly networks among authors in Iraq. The methodology encompasses a sequence of defined stages, each contributing to the holistic understanding of the academic scene. The architectural design commences with the accurate acquisition of data pertinent to authors in Iraq. To accomplish this, meticulously designed scrapers were employed, operating within the bounds of ethical data retrieval practices. The utilization of Google Scholar's API ensures that data retrieval is not only efficient but also respectful of the platform's terms of service. Google Scholar serves as the primary source, and to expedite this process, a strategy involving three virtual machines (VMs) was dynamically executed. Each VM was assigned a specific domain to navigate, enabling the rapid extraction of essential data. This parallelized approach was implemented to reduce the amount of time required for data aggregation.

To ensure the coherent dynamic storage of the collected data, a dedicated cluster has been created within MongoDB Atlas, a robust platform recognized for its durability and scalability. The subsequent phase is characterized by processing of the collected data. Amazon Web Services (AWS) provides the backdrop for this crucial stage. An accurate structured cloud cluster is created, comprising a Master Node and three worker nodes. These components, fortified with the strength of Apache Spark, collectively link the complex structure of data processing. Seamless access to the data residing within MongoDB Atlas is

facilitated through a specialized connector, Spark MongoDB Connector (SDBC). The inherent capabilities of Spark SQL library are fulfilled and used for data processing. After the processing procedures, the processed data is returned back into the MongoDB Atlas cluster. This accurate cycle ensures the constant availability of the processed data, a crucial element for final analyses.

The processed data lays the foundation for the creation of two proposed network models: the first being the Iraqi Scholar Network model (ISN), and the second proposed as the Iraqi University Network model (IUN). The former model seeks to map the complex tapestry of collaborative links between researchers in Iraq. Nodes within this network represent individual researchers, while the edges symbolize collaborative efforts, shared research interests, and co-authored works. The latter model casts a spotlight on the interweaving collaborations among Iraqi universities, highlighting the interaction between knowledge exchange and collaborative research between academic institutions. The penultimate phase embraces visualization through the utilization of Cytoscape software. This visualization breathes life into the network models, rendering them comprehensible and accessible. Additionally, the proposed ACDN serves as a dynamic metric to calculate the academic influence of individual Iraqi researchers and universities. This metric stands alongside specified metrics such as QS Rank and Webometrics Rank, facilitating a comprehensive evaluation of academic impact within the Iraqi scholarly landscape. The proposed architectural system demonstrated above is depicted in the following Figure 4.1.

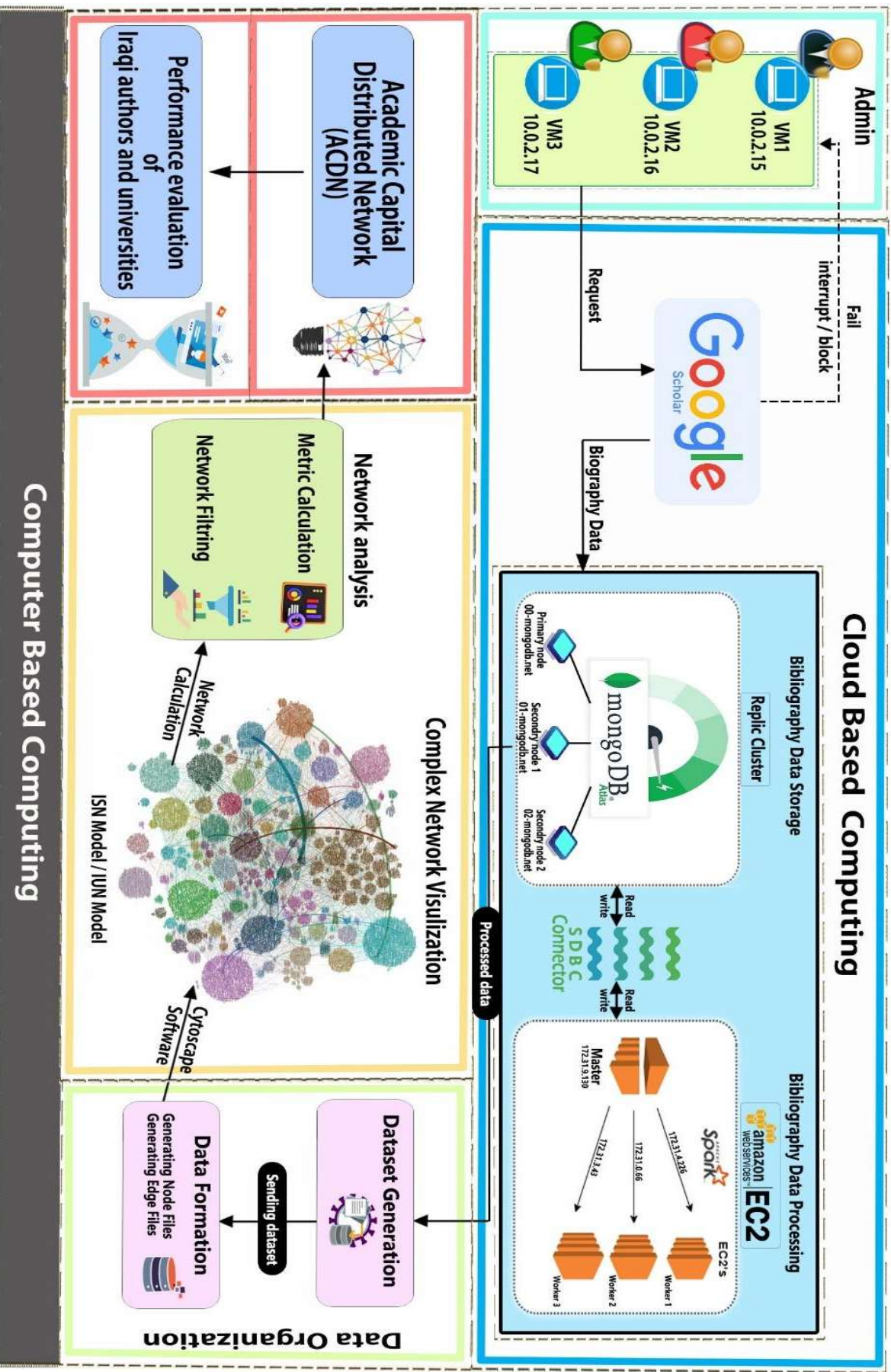


Figure 4.1: The proposed System Architecture

4.3 System Strategy

The ACDN Proposed System unfolds through a methodical approach, seamlessly navigating through key stages. Beginning with data collection, the proposed Unischolar Model and ScholarSync IRAQ Model play pivotal roles in acquiring diverse and comprehensive data from scholarly sources, setting the groundwork for subsequent analyses. Transitioning to the processing phase, data undergoes meticulous storage, organization, and deployment within dedicated clusters, fostering an organized structure. The journey continues with data transmission after preliminary processing and the deployment of data parallel processing through spark supported with AWS cloud cluster, ensuring efficiency and expediting the complex process of data analysis.

The subsequent phase focuses on data set formation, where the ACDN proposed system crafts two crucial network models – the Iraqi Scholar Network Model (ISN) meticulously maps collaborative links among researchers, capturing the intricate tapestry of scholarly connections, while the Iraqi University Network Model (IUN) sheds light on collaborative efforts and knowledge exchange within Iraqi academic institutions. The system reaches to the stage of network visualization, employing advanced techniques that animate the Iraqi Scholar Network Model (ISN) and the Iraqi University Network Model (IUN), rendering intricate relationships which are intelligible and accessible.

Finally, ACDN emerges as the pinnacle of the system, serving as a comprehensive metric to gauge the academic influence of individual Iraqi researchers and universities, enriching the understanding of scholarly dynamics in Iraq. This system architecture intertwines data acquisition, Storage, processing and parallelism use of each stage, network modeling, visualization, and impact assessment. This total approach reveals the intricate scholarly structure within Iraq, shedding light on the collaborative endeavors and academic influence shaping the intellectual structure. Figure 4.2 deals with the System work strategy diagram completely by considering all the steps that are described later in this chapter.

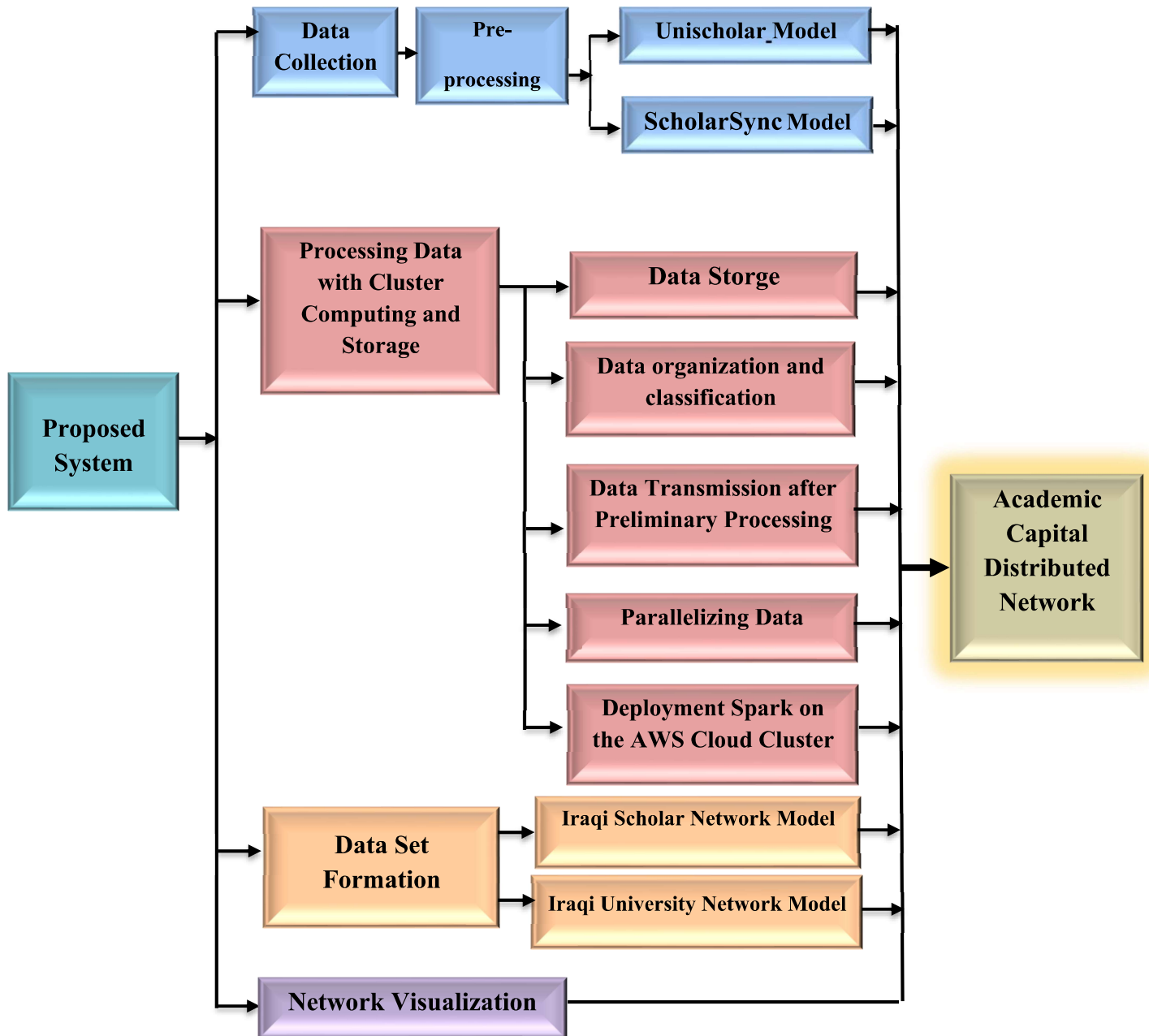


Figure 4.2: System Work Strategy Diagram

4.4 ACDN Proposed System

The ACDN proposed system introduces a comprehensive approach to data collection for academic research analysis. Leveraging the wealth of information available on Google Scholar profiles, the system aims to gather valuable insights by aggregating authors, publishing data, and associated keywords. Education Data Mining principles underlie in this direction, wherein a developed web scraper has been designed to accurately gather relevant online

links and extract essential data. The meticulous data collection process is guided by established parameters to ensure data quality, with two proposed models, Unischolar and ScholarSync used to facilitate efficient data retrieval and storage. Followed by a phase organization, storage and processing by taking advantage of the use of parallelism at each stage and then analyzing this data and visualizing it and proposing criteria for building this metric and making an evaluation of it. The following sections explain these models and highlight mechanisms and contributions to this work as follows:

4.4.1 Data Collection

The profiles that are seen on Google Scholar provide a significant amount of information regarding academic research. Such information includes authors, publishing data, and keywords that are related with each work. Google Scholar is a fast and user-friendly service that enables users to do in-depth searches for scholarly publications.

The objective of the field known as Education Data Mining is to provide resources for the investigation of particular data types that are generated in an educational setting. The web scraper collects all of the required online links, extract all of the required data from the source links, and save the data in a csv file. When data from Google Scholar profiles and Google Scholar articles are combined, the resulting system may be used for academic prediction. This allows for the creation of a powerful tool for assessing the environment in which academic research is conducted and the discovery of interesting ideas. To ensure that the collected data is of high quality and accuracy, several parameters are set up to control the crawling process. For example, the number of requests per minute is limited to avoid overloading the Google Scholar server and to implement a delay between each request. Two models are proposed to fetch this data and they are explained as follows:

4.4.1.1 Unischolar Model

The collection of important author data from Google Scholar has been made possible by the creation of a scraper using a smooth interaction between SerpApi and Scholarly. This potent combination uses SerpApi to explore academics' personal sites and acquire in-depth information, and scholarly sites to input university names and extract distinctive identifiers. The scraper seamlessly navigates through the enormous Google Scholar collection, offering

a quick and trustworthy way to retrieve and arrange author data for additional research and insights. This model is implemented by suggesting the three following scrapers:

Scraper1: (Search Organization)

The goal of this Scraper is to retrieve the IDs of all Iraqi universities listed on Google Scholar by constructing a web scraper utilizing the Scholarly library. The name of an Iraqi university is used as the first part of input for the algorithm, and this information (names of Iraqi universities) is obtained from the website of the Iraqi Ministry of Higher Education and Scientific Research. The algorithm focuses on extracting identifiers that are unique to each organization and which have a specific identifier. This is carried out for each university on Google Scholar, and the identities serve as individual and distinct identifier for each and every educational institution.

The approach that has been described, is referred to as algorithm 4.1. that is illustrates the methodical procedure of completing this work:

Algorithm 4.1 Fetching Researcher IDs from Scholarly
Input: CSV File with Universities Names Output: CSV File with Universities IDs
1. Start 2. Read the CSV file containing all universities names. 3. Clean the CSV file from any null or unwanted characters. 4. Create an empty DataFrame df2 with a column 'id' to store the extracted IDs. 5. Iterate over each row in the CSV file a) Retrieve the university name from the current row. b) Search for the university using the Search_org function from the Scholarly library. c) Convert the search result into a string representation. d) If the search result is empty, set ID as "00000000000000000000". e) If the search result is not empty, extract the ID from the search result using regular expressions. f) Remove any non-numeric characters from the extracted ID. g) Append a '\$' symbol to the end of the extracted ID. h) Assign the value of the 'id' column in the DataFrame df2 at the corresponding row index.

6. Merge the 'id' column from DataFrame **df2** to the original DataFrame **df** using the common row index.
7. Save the merged DataFrame **df** to a CSV file " Universities ID.csv".
8. End

Scraper2: (Search Author by organization)

The university ID, which was received from first scraper, must be input into scraper2 in order to retrieve authors information.

The second scraper successfully receives and gathers extensive information about researchers from universities by employing the "Search.author by organization" feature, and combining the acquired university IDs. This amount of data can be extremely helpful for academic research, collaboration prospects, and receiving insights into these universities' scholarly environments. The approach that has been described, is referred to as algorithm 4.2. that is illustrates the methodical procedure completing this work.

Algorithm 4.2 Fetching Author Information from Scholarly by Organization

Input: Universities IDs

Output: Text File with Author Information

1. Start
2. Import the necessary libraries (e.g., **scholarly**).
3. Use the **search_author_by_organization** function from the Scholarly library to search for authors based on the university IDs.
4. Open a text file "reportnew.txt" to store the fetched author information.
5. Iterate over the desired number of authors.
 - a) *Retrieve the next author from the search query using next (**search_query**).*
 - b) *Convert the author information to a string representation.*
 - c) *Write the author information to the text file.*
 - d) *Print the current iteration number for progress tracking.*
6. End

Scraper3: (Google-Scholar-Author)

The purpose of the third scraper, is to make use of the author ID that is obtained from scraper2 and retrieve author and article information from authors profile page on google scholar. The algorithm 4.3 below, illustrates the methodical approach of completing this work.

Algorithm 4.3 Scraping Scholarly Data (author and article information)

Input: CSV File with Author IDs

Output: MongoDB Database with Authors and Articles Information

1. Start
2. Import the necessary libraries (e.g., SerpApi, pandas, pymongo).
3. Connect to the MongoDB database using the provided connection string.
4. Read the CSV file containing the author IDs.
5. Clean the CSV file from any null or unwanted characters.
6. Create an empty DataFrame df to store the fetched article information.
7. Create an empty DataFrame df2 to store the fetched author information.
8. Open a text file "scraper33.txt" to store the fetched data (optional).
9. Iterate over the desired number of author IDs).
 - a) *Set the parameters (engine, author_id, num, Api_key) for the Google Scholar author search using the current author ID.*
 - b) *Perform the search using the SerpApi library and retrieve the search results.*
 - c) *Extract the required information from the search results, such as author details and article details.*
 - d) *Append the extracted author information to the DataFrame df2.*
 - e) *Append the extracted article information to the DataFrame df.*
 - f) *Write the extracted author and article information to the text file (optional).*
10. Save the DataFrame df to a CSV file "articles.csv" with UTF-8 encoding.
11. Save the DataFrame df2 to a CSV file "author.csv" with UTF-8 encoding.
12. Read the CSV files "articles.csv" and "author.csv" into DataFrames df and df2.
13. Convert the DataFrames df and df2 to documents.
14. Insert the dictionaries into their respective collections in the MongoDB database.
15. End.

Unischolar represents a comprehensive academic model, harnessing the combined capabilities of the Scholarly and SerpApi libraries. This multifaceted model unfolds in three distinct phases, commencing with a meticulous search for universities to establish a foundational understanding. It then seamlessly transitions to the identification and categorization of authors affiliated with these organizations. The endeavor reaches its zenith with an in-depth exploration of Google Scholar author profiles, ensuring a thorough and nuanced extraction of academic data. The Unischolar Model serves as a versatile and exhaustive tool, promising to unveil the intricate tapestry of scholarly connections and contributions. Figure 4.3 represents steps of Unischolar Model:

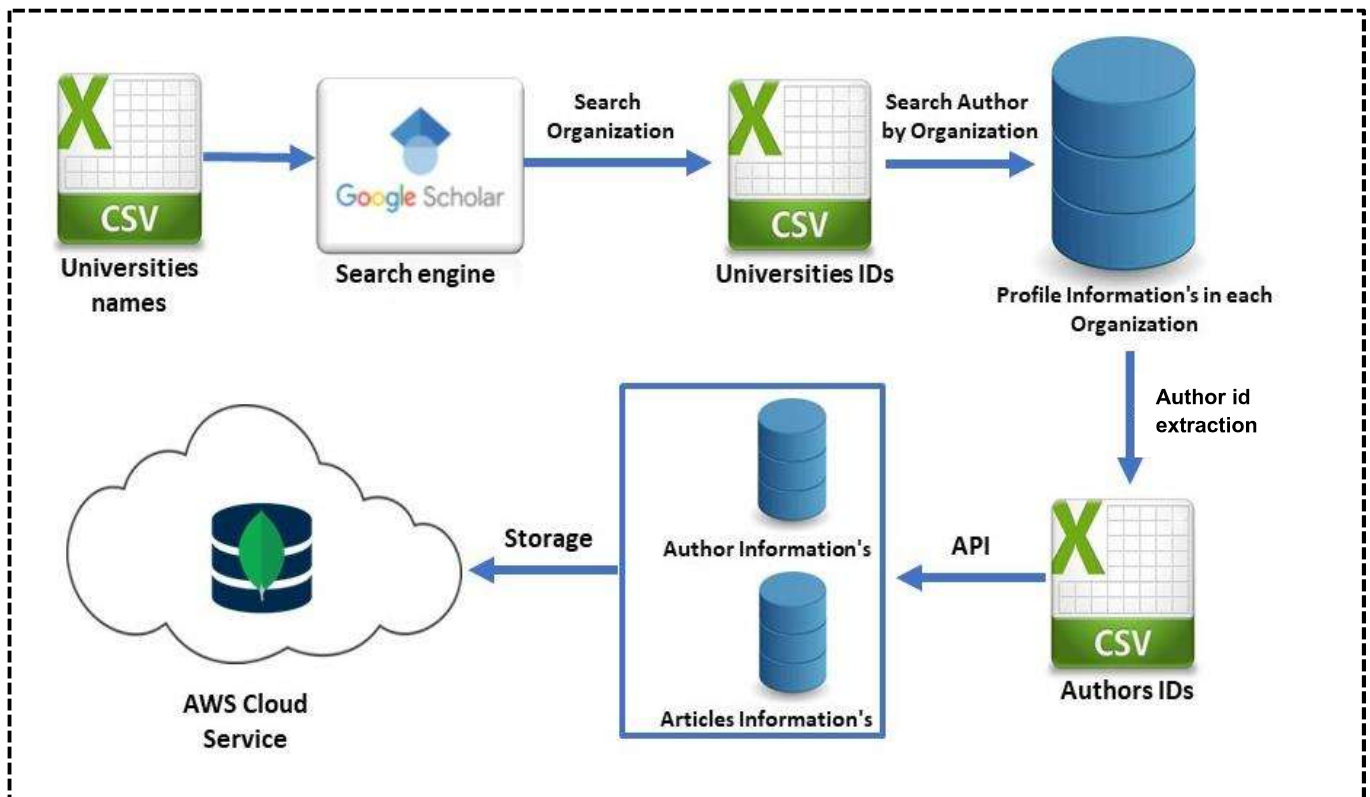


Figure 4.3 Steps of Unischolar Model

Some limitations were encountered, and these will be explained in Chapter 5. A different strategy was proposed to overcome these limitations by introducing and implementing a completely new technique illustrated in the ScholarSync IRAQ Model. A systematic process was provided by this approach to fill the gap and implement the ScholarSync IRAQ model successfully.

4.4.1.2 ScholarSync IRAQ Model

ScholarSync IRAQ is a robust methodology created to sync and collect research data from Iraqi academics and universities. ScholarSync IRAQ gathers a wide range of important data with its dual scraper system, which smoothly integrates simultaneous web scraping and focuses on author profiling. This methodology makes it simple to retrieve information about academics' biographies, affiliations, publications, citation metrics, and areas of interest quickly and effectively, providing a deep model of the academic environment at Iraqi universities. ScholarSync IRAQ enables authors, institutions, and administrators to obtain deep insights, foster collaborations, and make decisions that improve scholarly contributions of Iraqi academia by offering a consolidated and structured dataset. ScholarSync IRAQ Model provides rapid and precise retrieval of authors' data by employing SerpApi, offering a useful resource for research analysis. Researchers and analysts can streamline their data collecting with the help of scraper, freeing them up to concentrate on drawing relevant conclusions from the massive amount of data available on Google Scholar. After collecting the data, it is stored in the Cloud MongoDB Atlas. MongoDB Atlas is chosen because it provides high scalability, flexibility, and availability, which are crucial for managing large amounts of data. This model is implemented by suggesting two scrapers as follows:

Scraper1: (Google-Scholar-Profiles)

A strategy known as parallel web scraping, which makes use of many computers (virtual instances), has been put in order to collect data from Iraqi authors who post their scientific output on Google Scholar. There are three primary stages:

Step 1: Setting Up the Environment

In order to facilitate the scraping of multiple websites concurrently, multiple machines are used. This results in a total of three virtual machines (VMs) being created on the computer through Virtual Box's provision of three simulated instances of operating systems.

Step 2: Parallel Web Scraping

There are three Python scripts that are programmed to run concurrently on the three virtual machines. The first program is for "edu.iq," the second is for

"edu.krd," and the third is for "uod.ac". These scripts supported with SerpApi used in order to send many requests at the same time to Google Scholar. As a result of this parallelism, the speed and efficiency of web scraping are significantly improved, as the data of different authors can be extracted simultaneously.

Step 3: Extracting Information from Iraqi Researchers.

The goal is to gather information from scholars in Iraq who use official email addresses with specified domains, including "edu.iq," "edu.krd," and "uod.ac." These scripts for web scraping are meant to target the profiles of these scholars and collect crucial information, such as author names, affiliations, publication titles, h-indexes, i10 indices, citations, and et al.

Algorithm 4.4 demonstrates scraping Google Scholar profiles and saving results to the cloud.

Algorithm 4.4 Scraping Google Scholar Profiles and Saving Results

Input: available domains in Iraq universities (edu.iq, edu.krd, uod.ac)

Output: CSV File with Profile Information

1. Start
2. Import the necessary libraries (e.g., os, sys, time, SerpApi, urlsplit, parse_qs, pandas).
3. Open a text file "Profile Information text.txt" to store the extracted data.
4. Set the parameters for the Google Scholar profile search.
5. Perform the search using the SerpApi library and retrieve the search results.
6. Create an empty list `profile_results_data` to store the fetched profile information.
7. Set `i` to number of results in one page (10) as the starting value for indexing the search results.
8. Retrieve the search results and iterate over each profile in the search results.
 - a) *Extract the required profile information, such as thumbnail, name, link, author ID, affiliations, email, cited by, and interests.*
 - b) *Append the extracted profile information to the `profile_results_data` list.*
 - c) *Print the current progress of profile extraction.*
9. Create a DataFrame `pd2` using the `profile_results_data`.
10. Save the DataFrame `pd2` to a CSV file "Profile Information.csv" with UTF-8 encoding.

11. Check if there is a "next" pagination link in the search results.
 - a) If present, update the search parameters with the next page's parameters and increment i by 10.*
 - b) Repeat steps 5-11 for the next page of search results, until there are no more profiles to fetch.*
12. Print a message indicating that the results have been saved.
13. End

Scraper2: (Google-Scholar-Author)

The first web scraping procedure (scraper 1) collects all the data from the authors profile, and the author ID of the researcher is chosen to be used by the second web scraper which scrapes the researcher's personal page using SerpApi and the author ID it has previously obtained.

This information is generally referred to as "author information," and it consists of the researcher's affiliations, interest, and other essential particulars. Second scraper collects article information related to identified author. This scraper is able to interact with author results pages in a time-efficient manner and go through the researcher's profile completely due to limited number of results in each page, resulting in extraction and arrangement of the data in a systematic manner.

The proposed Algorithm 4.5 for extracting scientific data (author and article information) is as follows:

Algorithm 4.5 Scraping Scholarly Data (author and article information)
Input: CSV File with Author IDs Output: MongoDB Database with Authors and Articles Information
1. Start 2. Import the necessary libraries (e.g., SerpApi, pandas, pymongo). 3. Connect to the MongoDB database using the provided connection string. 4. Read the CSV file containing the author IDs. 5. Clean the CSV file from any null or unwanted characters. 6. Create an empty DataFrame df to store the fetched article information. 7. Create an empty DataFrame df2 to store the fetched author information. 8. Open a text file "author, article.txt" to store the fetched data (optional). 9. Iterate over the desired number of author IDs).

- a) Set the parameters (engine, author_id, num, Api_key) for the Google Scholar author search using the current author ID.*
 - b) Perform the search using the SerpApi library and retrieve the search results.*
 - c) Extract the required information from the search results, such as author details and article details.*
 - d) Append the extracted author information to the DataFrame df2.*
 - e) Append the extracted article information to the DataFrame df.*
 - f) Write the extracted author and article information to the text file (optional).*
10. Save the DataFrame df to a CSV file "articles.csv" with UTF-8 encoding.
 11. Save the DataFrame df2 to a CSV file "author.csv" with UTF-8 encoding.
 12. Read the CSV files "articles.csv" and "author.csv" into DataFrames df and df2.
 13. Convert the DataFrames df and df2 to dictionaries.
 14. Insert the dictionaries into their respective collections in the MongoDB database.
 15. End

ScholarSync IRAQ Model, represent a focused and efficient scholarly data extraction tool powered by SerpApi. This model is tailored to the specific needs of profiling and understanding Iraqi academia, honing in on Google Scholar profiles and authors. By utilizing the precision of SerpApi, it ensures a direct and effective retrieval of essential scholarly information related to Iraq. Whether scrutinizing individual Google Scholar profiles or tracing the academic contributions of specific authors within the Iraqi academic landscape, the ScholarSync IRAQ Model stands as a dedicated and streamlined solution for scholarly data extraction with a distinct emphasis on Iraq's scholarly environment. This model is illustrated in Figure 4.4.

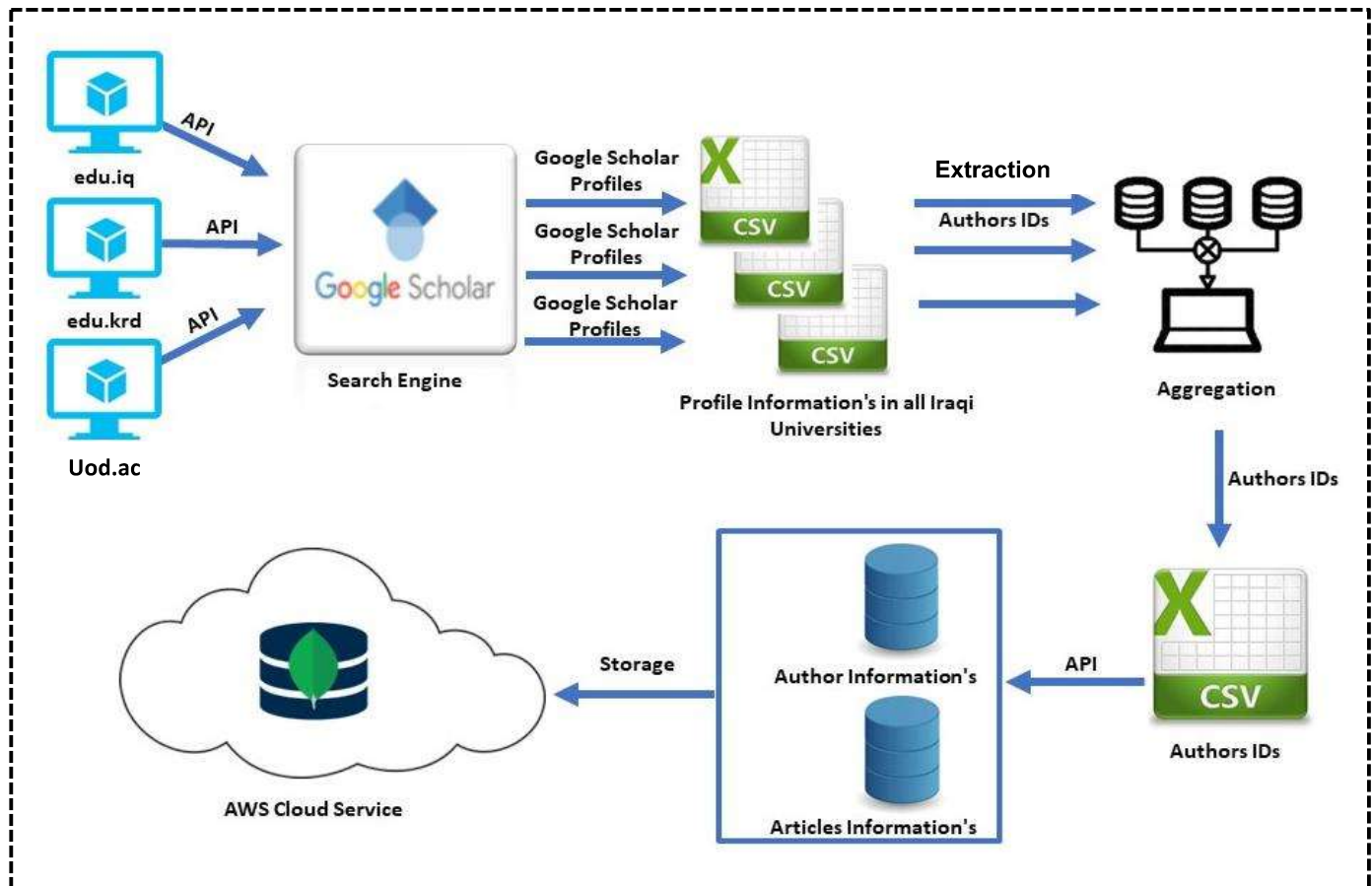


Figure 4.4 Steps of ScholarSync IRAQ Model

4.4.2 Processing Data with Cluster Computing and Storage

The authors data that generated from Google Scholar needs to be organized, that is, the process of storing and retrieving it needs to be as quick as possible. Since the system is time constrained and has a continuous data stream, the framework must be selected according to the principle of this restriction. The MongoDB database is used to store and retrieve data due to scalability and versatility in managing semi-structured and unstructured data. Cloud MongoDB Atlas is used as fully managed database service that makes it easy to deploy, manage, and scale MongoDB databases. In addition, Apache Spark, which specialises in parallel processing of data is also used. The MongoDB Atlas database is linked with Apache Spark via the Spark-MongoDB connector.

4.4.2.1 Data Storage

By using the distributed database platform Cloud MongoDB Atlas, a scalable data warehousing system is created to provide effective and dependable storage of data Scraped from Google Scholar, which includes all papers for each author, author information, citations, and associated metadata. The information is then carefully arranged and transformed into a structured format that can be stored and examined. Utilizing MongoDB Atlas Cloud's capacity to store data in a document-oriented manner is one of its key benefits. As a result, a flexible, schema-free data format is possible, as the scraped data is saved as JSON and CSV. Algorithm 4.6 shows the steps used to store the scraper results:

Algorithm 4.6 Storing Data Scraping in Cloud Distributed Database

Input: Website Links

Output: Distributed Database with Extracted Features

1. Start
2. Create a MongoDB Atlas account and set up a data cluster.
3. Install the pymongo package using pip and import the necessary libraries (e.g., pymongo).
4. Establish a connection to the MongoDB Atlas account using the pymongo driver.
5. Connect to the specific cluster within the account.
6. Retrieve the search results from a SerpApi search query and parse them into a Python dictionary format.
7. Create a new "collection" in the MongoDB database to hold the scraped data.
8. Iterate over the search results:
 - a) *Extract the desired features (e.g., researcher citation, affiliation, interest) from each search result.*
 - b) *Format the extracted features as required.*
 - c) *Create a new document in Python dictionary format to hold the extracted features.*
 - d) *Insert the document into the collection in the MongoDB database.*
9. Continue data scraping and archiving, ensuring that the extracted features are correctly formatted and inserted into the MongoDB database.
10. Use the pymongo driver to query the database and extract the appropriate fields from the documents to retrieve the stored data.
11. Process the retrieved data as needed (e.g., sorting, filtering).
12. End

4.4.2.2 Data organization and Database Deployments

The MongoDB Atlas database is used to organize and classify data due to its compatibility with spark framework. Also, the support provided by AWS Cloud service is used for deploying the generated database. A database has been created based on MongoDB Atlas, and it is described by specifying (database name, Collections name, username, password, Api-key, Cluster, provider/region, data Api access, whitelist IP and containing field) and containing fields, like (*thumbnail, name, link, author_id, email, affiliations, cited_by, interests, email, website, interests, cited-by-table, cited-by-graph, public-access-link, available-public-access, co-authors, not-available-public-access, name of author, article-title, article-link, article-citation-id, article-authors, article-publication, article-cited-by-value, article-cited-by-link, article-cited-by-cites-id*) Figure 4.5 shows the diagram of the created MongoDB Atlas database.

After completing the distributed database settings, A Cluster instance was created in MongoDB Atlas, which is a collection of servers that work together to store and manage data. These servers are distributed across multiple geographies to ensure high availability and low latency for data.

A cluster instance consists of a primary node and two secondary nodes. The primary node is responsible for handling all write operations, while the secondary nodes are responsible for replicating data from the primary node and handling read operations. If the primary node fails, one of the secondary nodes is automatically promoted to become the new primary node. This ensures that the application can continue to read and write data even in the event of a server failure. Some utilities and services have been used to manage and monitor a cluster instance, The monitoring of database performance and health is carried out using the MongoDB Atlas web console, while data exploration, along with query execution is performed using MongoDB Compass.

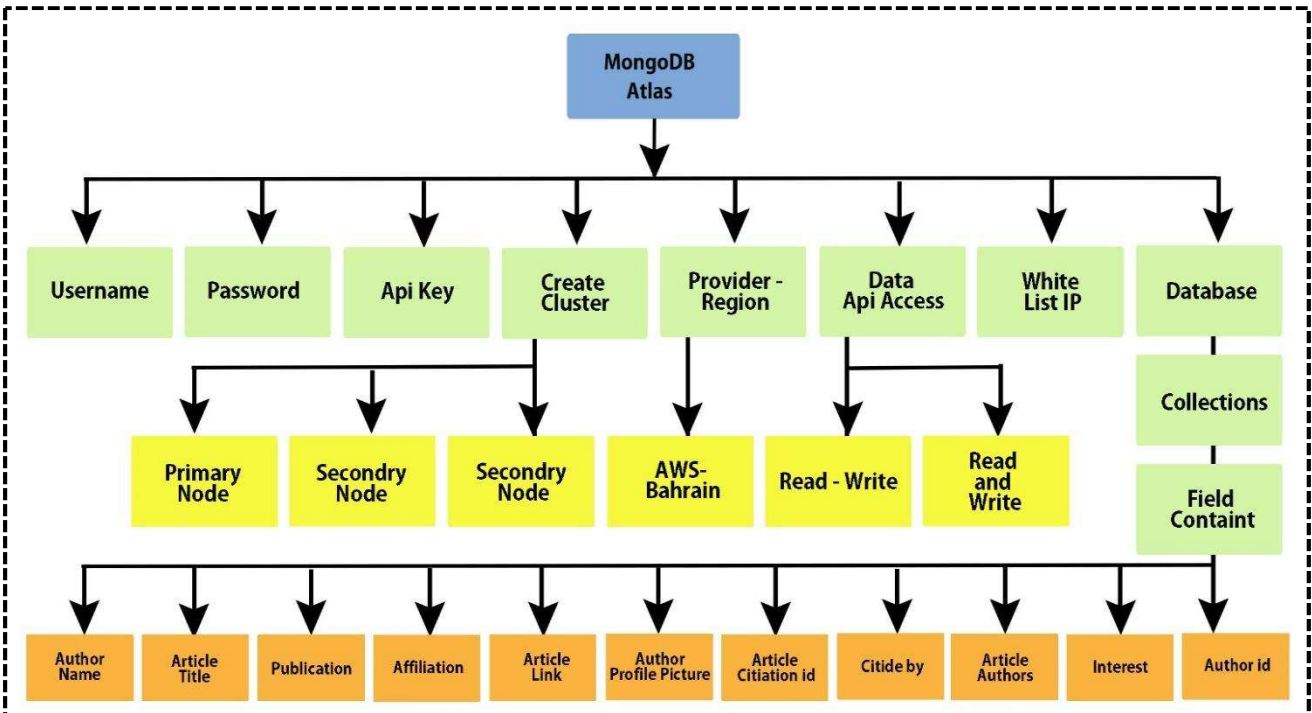


Figure 4.5 Deploying a Database in MongoDB Atlas

A database has been created, which may be thought of as containers for the various types of data stored in the database. Within the database, many collections have been created to organize and categorize the data stored in the database. A collection is a group of documents that are all related to one another. These documents were saved in a format known JSON. It is possible for each document in a collection to have its own distinct schema. This means that documents belonging to the same collection can have a variety of data fields and types.

4.4.2.3 Data Transmission after Preliminary Processing

Raw data needs to be organised, cleaned, and filtered. Therefore, big data frameworks must be used. One of the best big data frameworks is Apache Spark. Apache Spark can handle data in parallel form and is considered very quick in processing large datasets. The Apache Spark and MongoDB Atlas systems can be integrated into one another with the help of a tool called Spark MongoDB Connector (SDBC). This connector is an official connector that provides a seamless integration between Spark and MongoDB. It allows to read and write data from MongoDB collections as DataFrame objects in Spark. This enables the use of the complex processing capabilities of Spark in order to examine and

manipulate the data that is kept in MongoDB. Figure 4.6 shows the integration between MongoDB and Apache Spark.

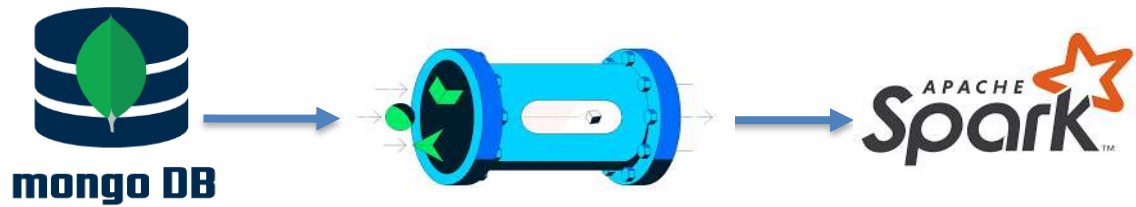


Figure 4.6 Integration Between MongoDB and Apache Spark

The latest version (6.0.9) of MongoDB is used, and it is supported by the connector (v10.2), which enables a variety of actions, including receiving data from MongoDB, filtering and aggregating data within Spark, and publishing data back to MongoDB. In addition, the connection provides support for DataFrame-based API in Spark. For the sake of structured storage and expedient retrieval, the data obtained from Google Scholar has been saved in a MongoDB Atlas.

4.4.2.4 Data Parallel Processing with Apache Spark

Authors data, which is obtained through a scraper from Google Scholar, has been collected and deployed in MongoDB Atlas. Subsequently, a cluster environment using four simulated EC2 servers is created on the AWS Cloud, consisting of a Master node and three Worker Nodes. In order to overcome the difficulties posed by this data which may include inconsistencies, incomplete information, and potential variations in data formats, Spark Framework is installed on the cluster nodes. By utilizing the powerful capabilities of Spark, the data can be processed and various issues can be resolved within the dataset. To create Apache spark cluster, three worker nodes and one master node, which serves as the cluster management, make up the cluster architecture as shown in Figure 4.7.

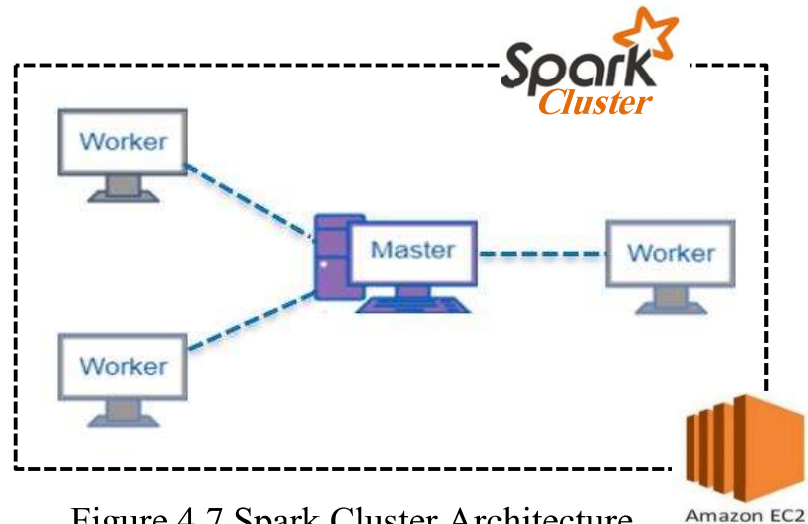


Figure 4.7 Spark Cluster Architecture

On master node, a single worker is always active by default. The parameters, number of nodes and machine type are used to specify the cluster resources. The master process schedules the worker resources (available CPU cores and memory) after accepting them for execution. Worker nodes are responsible for executing tasks assigned within the job. The master node terminal interface is linked to master node to run it, which also runs Apache Spark and JupyterLab, was used to submit applications. Finally, the Spark driver is the application that connects to the master node and creates the SparkContext. Through the setup of an Apache Spark Cluster, including several slave (worker) nodes and a master node, the configuration of Apache Spark Core is explained in Figure 4.8.

The fundamental framework utilized in Apache Spark is known as Spark Core. This framework renders its elements and functionalities as accessible through Python programming language APIs. When a script is executed in PySpark, it gives rise to the creation of RDDs (Resilient Distributed Datasets), enabling Spark to execute parallel data processing on a cluster.

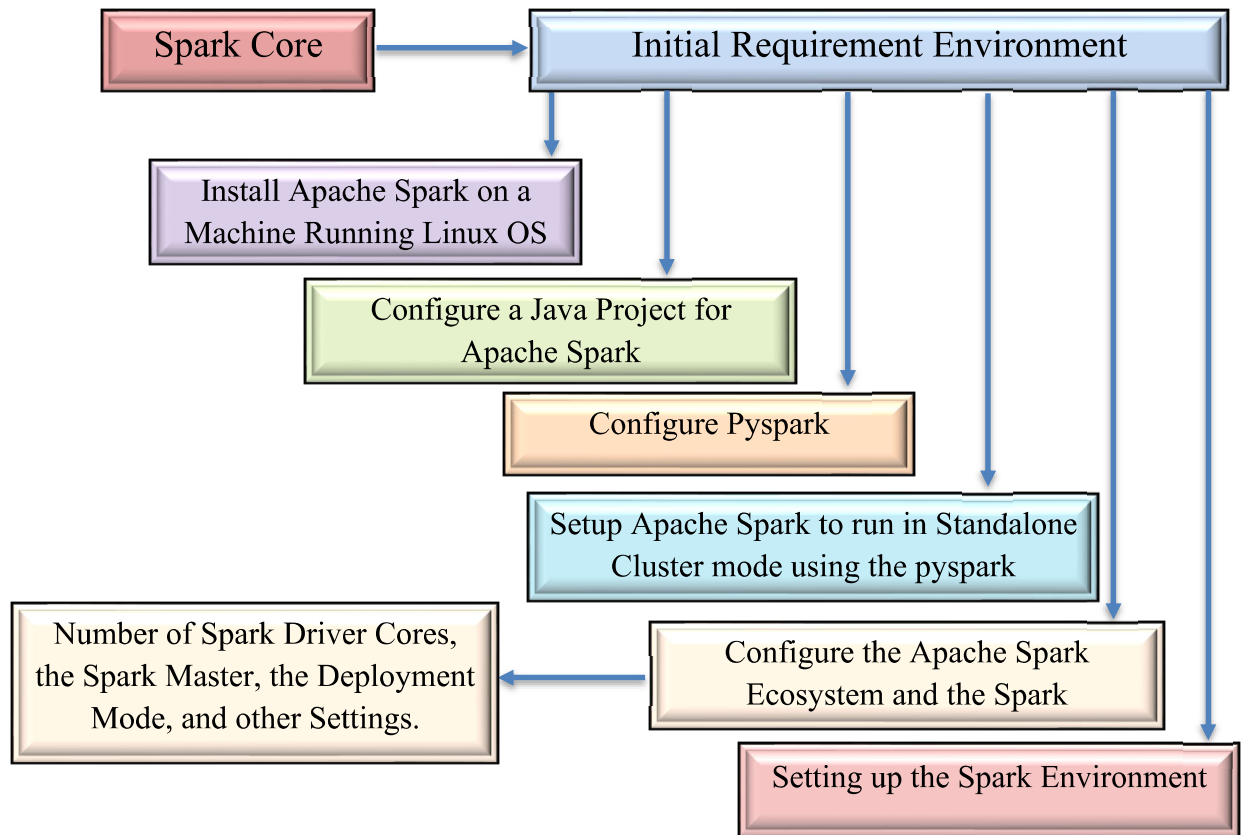


Figure 4.8 Spark Core Configuration

In the data processing workflow within Apache Spark, researchers' data from Google Scholar is efficiently handled within the cluster environment. The process begins with the retrieval of this valuable information from MongoDB Atlas, a repository where the researchers' data is securely stored. This initial step ensures that the data is accessible securely. Upon successful data retrieval, the information is organized and structured within a Spark DataFrame, a powerful and flexible data structure in Spark. This DataFrame serves as the foundation for conducting a series of operations tailored to extract meaningful knowledge from the researchers' data. The subsequent stages of the processing encompass a range of operations to process scholar data as shown in Algorithm 4.7.

Algorithm 4.7: Data Processing for Scholar Data Using Apatch Spark

Input: Scholar Data CSV, MongoDB Atlas Documents

Output: Processed Data Stored in MongoDB Atlas

1. Start
2. Initialize Spark session
3. Creating a DataFrame
4. Data Preprocessing
 - Data Ingestion: Read data from MongoDB Atlas into Spark DataFrames
 - Data Transformation and Data Cleaning (Filter, Map, Join, Aggregation, Sorting)
5. Perform Data Processing
 - Drop column.
 - Handle missing values.
 - Remove duplicates.
 - Convert data types.
6. Connect to MongoDB Atlas
 - Establish a connection to MongoDB Atlas using Documents.
7. Create a new MongoDB database
8. Define a Collection to Store Processed Data
9. Convert Spark DataFrame to Pandas DataFrame
10. Insert the processed data into the MongoDB collection.
11. Close MongoDB Client Connection
12. End

4.4.2.5 Deployment of Spark on AWS EC2 Cloud Cluster

To deployment Spark on the AWS Cloud Cluster, the following configuration is needed:

4.4.2.5.1 Apache Spark Ecosystem Configuration

Some parameters, such as the number of Cluster nodes, the number of cores on each node, the available memory on each node, the launchable number of threads, the deployment mode, additional Java options, and additional library paths, depend on the cluster configuration or the needs of users. Control over these parameters is bestowed upon us as users of the Apache Spark application,

allowing the customization of the Apache Spark ecosystem to align with the requirements of application. The parameters for configuring the Apache Spark Ecosystem are illustrated in Figure 4.9.

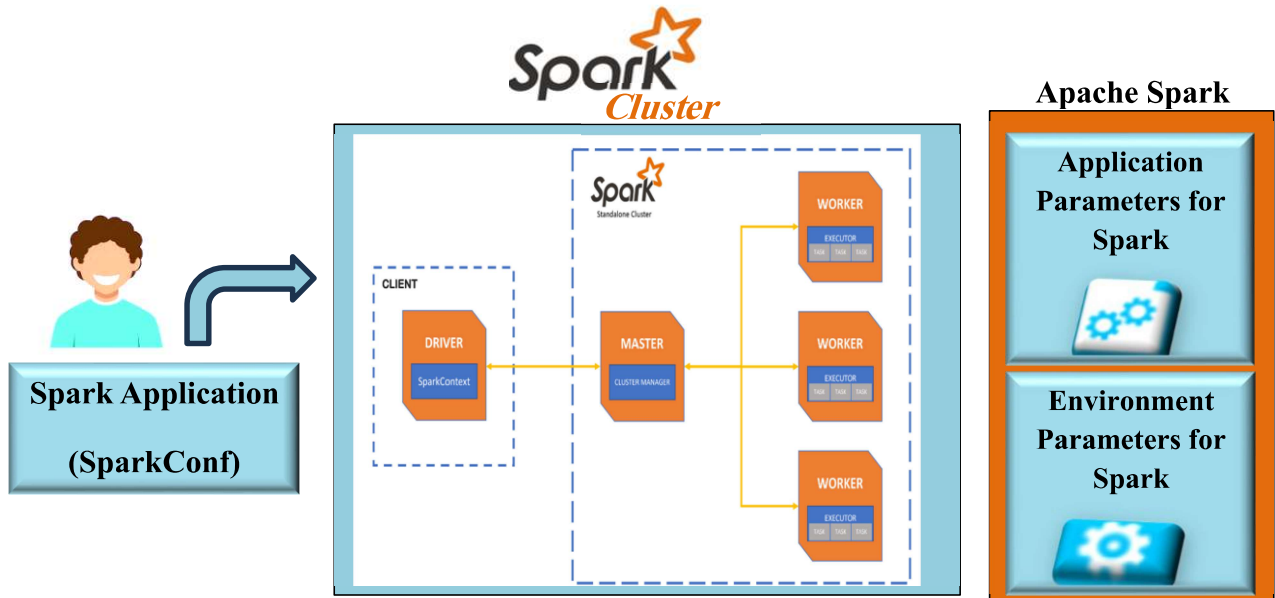


Figure 4.9 Parameters for Configuring Apache Spark Ecosystem

The two main types of settings that are configured to enable the customization of the Apache Spark ecosystem are shown below.

- **Application Parameters for Spark**

By using the SparkConf object in the driver, Spark Application Parameters were set up in the Spark application itself. These parameters are illustrated in Figure 4.10. like key-value pairs, Spark application parameters are configured using SparkConf. The spark application is sent to the command prompt using *spark-submit* command.

```
SparkConf sparkConf = new SparkConf().setAppName( "Test4444 application" )
    .setMaster( "Cluster" )
    .set( "Spark.executor.memory","2g" );
```

Figure 4.10 Configuration of Spark Parameters

- **Environment Parameters for Spark**

These variables influence how the Cluster's nodes act and how much memory they use. The *spark-env.sh* shell script's environment variables must be set up manually for each node in the spark Cluster. Apache Spark is comprised of four libraries, and the one used here is "Spark SQL library". This library enables Apache Spark to access structured, semi-structured, as well as unstructured data, and optimizes operations on that data using Spark SQL. The illustration in Figure 4.11 depicts the architecture that has been used which includes the data source, Dataset, DataFrame, and Spark SQL.

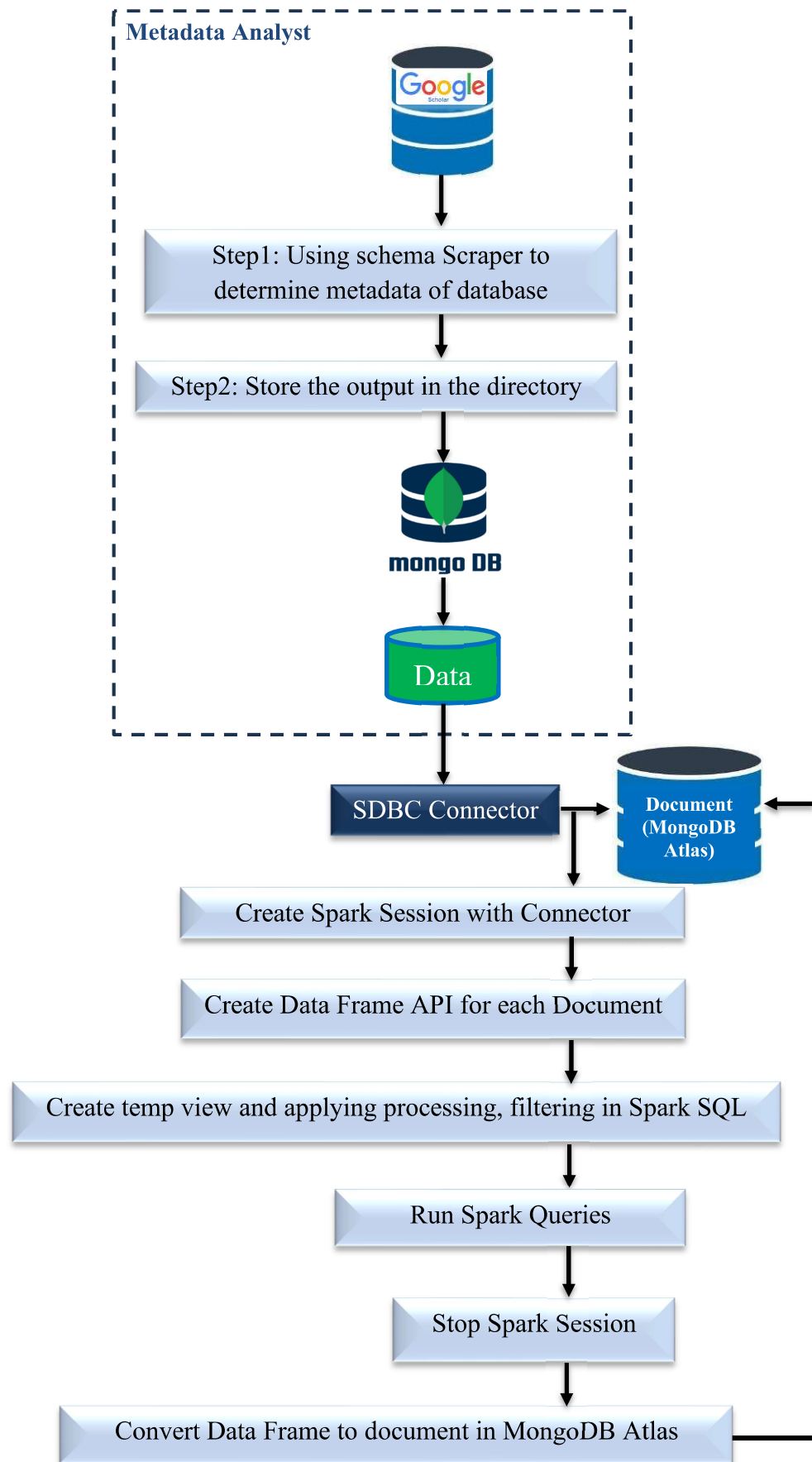


Figure 4.11 Architecture Based in Apache Spark Layer

4.4.3 Adopting Parallelism in Proposed System

In this work, parallelism have been achieved at various stages, including: web scraping process, data storage, and data processing. The parallelism achieved in each component is discussed as follows:

1. Web Scraping:

Three virtual machines (VMs) are operated as a distinct instance of VirtualBox. Each VM is tasked with data retrieval from a specific domain which explains parallelism. All three VMs collaborate in parallel to concurrently collect data from their respective domains. Each VM functions autonomously and concurrently, so the parallelism here revolves around VMs functioning simultaneously.

2. Data Storage:

Within MongoDB Atlas cloud environment, a cluster of three nodes of MongoDB replica set type is configured, which facilitates data distribution across multiple nodes for redundancy and scalability purposes. Data storage based on a replica set includes parallelism, meaning that data can be written to and read from privileged nodes concurrently thus, improving read and write throughput and ensuring fault tolerance. This form of parallelism transpires within the database system, concentrating on load distribution and data availability.

3. Data Processing:

The existing Spark distributed system in this work includes a master node and three worker nodes. When data is imported from MongoDB Atlas into Spark, Spark's parallel processing capabilities are enabled. Data processing tasks are divided into smaller units referred to as partitions, and these partitions can be processed in parallel across worker nodes. Spark coordinates distributed processing by executing tasks on worker nodes at the same time, thus, speeding up data processing overall. Spark facilitates parallel data processing across diverse worker nodes. The data is partitioned, with each worker node autonomously processing its designated partition. This form of parallelism operates within the framework of data processing.

The "Bernstein's Law of Parallelism" was implicitly relied upon in this work, to accelerate web scraping, this approach aligns with Bernstein's Law, as it exploits parallelism by distributing tasks across multiple instances to improve overall efficiency. In data storage, this form of parallelism aligns with Bernstein's Law, as it leverages parallel processing within the database system to optimize load distribution and ensure data availability and data processing. With a master node and three worker nodes, Spark partitions data and executes tasks simultaneously across nodes, maximizing parallelism. Each worker node independently processes its designated data partition, ensuring efficient distributed processing. Overall, the utilization of parallelism in Spark, adheres to Bernstein's Law by effectively breaking down tasks into parallelizable units, coordinating simultaneous execution across distributed nodes, and enabling autonomous processing to accelerate data retrieval, processing, and storage.

4.4.4 Adopting Virtualization in Proposed System

Virtualization plays a crucial role in the efficient execution of the data scraping process for the utilization of Google Scholar. *In the first scenario*, the entire system leverages virtual machines (VMs) through VirtualBox, providing a robust platform for isolating and managing multiple scraper programs. Virtualization involves the creation of simulated computing environments within a single physical system, allowing the concurrent operation of multiple operating systems or applications on the same hardware. In this context, VirtualBox serves as the hypervisor, enabling the deployment of distinct VMs on a host machine. Each scraper program operates within its dedicated virtual environment, offering isolation from the underlying hardware and other VMs. This isolation ensures that the activities within one VM do not interfere with the operation of others, enhancing, in turn, the overall stability and security of the system.

The benefits of virtualization in this scenario are manifold. Firstly, it provides flexibility, as VMs can be easily created, duplicated, or modified to accommodate different scraping requirements. Various configurations can be experimented without affecting the host machine or other virtual instances. Additionally, virtualization promotes resource efficiency, enabling optimal utilization of the host machine's capabilities by dynamically allocating and managing resources such as CPU, memory, and storage for each VM.

Furthermore, the architecture of virtualization enhances scalability. As the demand for data scraping increases, additional VMs can be deployed to distribute the workload effectively. This scalability ensures that the system can adapt to handle larger datasets or more complex scraping tasks without the need for extensive hardware modifications. In summary, virtualization, facilitated by VirtualBox, plays a pivotal role in creating flexible, efficient, and scalable environment for data scraping activities. The use of VMs allows for isolated execution of scraper programs, which optimize resource utilization and provide scalable solution to meet evolving needs.

In the second scenario, virtual simulation conducted on AWS, that is, EC2 instances plays a pivotal role in orchestrating a scalable and efficient infrastructure for the processing of researchers' data from Google Scholar. Utilizing EC2 instances allows for the creation of a cluster comprising a master and three worker nodes, resulting in an established distributed computing environment. The EC2 instances serve as the virtual servers, where the extracted data stored in MongoDB Atlas are processed within distributed environments in the cloud, promoting data processing in dynamic and parallelized form. The benefits of EC2 in this context lie in its ability to scale resources based on computational demands, ensuring optimal performance during data processing tasks. The architecture of the EC2-based setup includes a master node and three worker nodes, where EC2 instances facilitate parallelized data processing which optimize the overall workflow.

The deployment of Apache Spark on these EC2 instances further enhance the computational capabilities, enabling distributed data processing across the cluster. The master and worker nodes, collaborate harmoniously to process the extracted data stored in MongoDB Atlas, leveraging the cloud's storage capabilities. The connectivity between Apache Spark and MongoDB Atlas is established through the Spark-MongoDB connector, ensuring a smooth and efficient exchange of data.

4.4.5 Data set formation

After collecting the Iraqi authors' data from the GS, a two-file dataset is generated the two files are represented as node and edge, and they are explained as follow:

A **node file** contains the number of nodes in a network, a node represents an individual or an entity, and the number of nodes in a network represents the total number of people or organizations connected through that network. The number of nodes in a network is an important factor in determining its complexity and connectivity between nodes. The **edge file**, however represents the total number of connections in the network which is equal to the number of edges in that network. An edge refers to a link between two nodes, and the number of edges represents the total number of connections in the network. The number of edges in the network reflects the level of interaction or connectivity between nodes. A network with multiple edges may indicate a high degree of cooperation or exchange of information between nodes. Algorithm 4.8 shows the network generation of the co-authors' data to the researchers.

Algorithm 4.8 Network Generation for Researcher Co-authors Data

Input: Researchers Names and Coauthors Data

Output: CSV File with Co-author Network Information

1. Start
2. Import libraries: Pandas, SerpApi (Google Search)
3. Open-source CSV files contain author and coauthor names as rows for each paper and are saved in a DataFrame (df).
4. Open another CSV file that contains author and coauthor names with an ID for each one, and save it in another DataFrame (df3).
5. Create a new empty DataFrame (df2) with the column's 'index', 'source', 'target', and weight.
6. Loop through each row and column of df:
 - a. Check if the value in the current column exists in the 'coauthor names' column of df2.
 - i. *If yes, get the index of that row and increment the 'weight' value by 1.*
 - ii. *If no, add a new row to df2.*
 - b. Get the index of the row in df3 where the 'names' column matches the value in column 'name' for the current row in df.

- c. Set the 'source' column and 'target' column of the new row to the 'index' value of that row in (df3), with the weight' column of the new row set to 1.
- 7. Write (df2) to the output CSV file in UTF-8 encoding.
- 8. Print "Done..." to indicate the completion of the program.
- 9. End

Two proposed models have been configured in this work, and they are described as follows:

4.4.5.1 Iraqi Scholar Network Model (ISN)

The first model includes a nodes file that represents information Iraqi researchers, that is, each node represents an author, the edge file, however, represents the researchers participating in the research, that is, the edge is formed if there is joint research between them, and the weight of the edge between them increases according to the number of papers associated with the two researchers.

4.4.5.2 Iraqi University Network Model (IUN)

The second model represents nodes file information about the collected universities and their characteristics. Edge file consists of interuniversity relationships. The strategy of creating edges between universities relies on co-authoring research papers between universities. There is an advantage between two universities if their authors co-author a research paper or collaborate on a research paper (two authors from different universities have a joint research paper). The weight of the edge between the two universities is increased according to the number of research paper attached to the two universities.

4.4.6 Network Visualization

After processing and downloading data from MongoDB Atlas cloud, a dataset is created from two files: node files and edge file. The node file includes information about the collected authors or universities and their attributes. The edge file consists of the relationships among the authors or universities of the same and different universities. The relations among the Iraqi universities are extracted using the nodes attributes, for example, when two authors are connected, if the authors are affiliated to different universities, then an edge is

created between the two universities. Thus, the network of authors represents the “Coauthorship Network” and the network of universities represents the “Universities Network”. Therefore, the strategy of creating edges among universities is based on co-authoring papers between the universities’ authors. An edge is made between two universities if their authors co-authored/collaborated on a paper. The weight of an edge between two universities is increased according to the number of papers associated with both. These complex connections are meticulously woven through a strategic thread of co-authored research papers, supported by the utilization of the Cytoscape software. As nodes and edges converge, and the weight of connections intensifies, Cytoscape empowers researchers, not only to watch the networks, but also understand the collaborative knowledge creation between Iraqi scholars and universities.

4.5 Academic Capital Distributed Network (ACDN)

The term (ACDN) refers to the value that can be placed on an individual's or institution's reputation in the academic world, due to their level of academic success. It takes a variety of factors of academic success, such as the quantity of publications, the number of citations, and the effect made in the relevant academic field. The criteria that have been proposed for building academic capital distributed network are:

1. Number of Papers (P)

The total number of articles that an individual or institution has contributed to academic journals or conference proceedings is denoted by the variable "P", which refer to "Number of Papers." It is a quantifiable measurement of the amount of academic production that can be achieved by an individual or institution.

2. Citations (C)

The variable "C" represents the count of instances where the research output of an individual or institution has been referenced by other researchers. Citations serve as a quantitative metric assessing the impact of an individual's or institution's scholarly work within the academic community.

3. H-Index (HI)

The H-Index, known as HI, is a metric that seeks to assess the influence and productivity of a researcher's work. It is defined as the number of publications that a single author has published (n) and which have received at least (n) citations.

4. I10-Index (H10)

The H10-Index (H10) is quite Similar to the H-Index; however, it only takes into account the individual's 10 most-cited publications. It is a measure of effect and production that is more selective in nature.

5. Eigen Centrality (E)

Author or institution's Eigen Centrality (abbreviated as E) is a measurement of how important they are to a network of academic collaboration. Not only does it take into consideration the quantity of connections author or institution has, but it also takes into account the quality of those connections.

6. Clustering Coefficient (CO)

The Clustering Coefficient (CO) is a measurement of the density of links among the author who collaborate with a certain individual or institution. It determines how closely an individual's or institution's network of collaborators is knit together.

7. Degree Centrality (D)

Author's or an institution's degree of centrality, which is denoted by the variable D, is determined by the number of links they have in a network of academic collaboration. It accomplishes this by counting the number of connections that an author or organization has with other research institutions.

8. Closeness Centrality (CL)

The Closeness Centrality (CL) index calculates the degree to which an individual or institution is connected to other scholars who are participating in an academic collaboration network. It is a measurement of how easy an author or institution can communicate with other researchers.

9. Betweenness Centrality (B)

In a network of academic collaboration, the betweenness centrality (B) index assesses the extent to which author or institution functions as a mediator between other scholars. It is a measurement of the relevance of an individual or institution in fostering communication and collaboration with other researchers.

The next step of the process of calculating the academic capital is to perform the weights assignment process. This process assigns a particular weight to each of the aforementioned factors. The assignment of weights is based on two aspects; *First*, experts are consulted about the most adequate values that fit the purpose of this thesis, and provide an accurate evaluation of the status of an author or institution. *Second*, several tests are performed on the most appropriate values of weights. As a result, the following weight values are considered in calculating the academic capital of authors and universities:

The weight of the *number of papers* P is:

$$W_P = 0.2 \quad (4.1)$$

The weight of the *citations* (C) is:

$$W_C = 0.2 \quad (4.2)$$

The weight of the *H-Index* (HI) is:

$$W_{HI} = 0.05 \quad (4.3)$$

The weight of the *H10Index* (H10) is:

$$W_{H10} = 0.05 \quad (4.4)$$

The weight of the *eigen centrality* (E) of an author or a university is:

$$W_E = 0.1 \quad (4.5)$$

The weight of the *clustering coefficient* (CO) of an author or a university is:

$$W_{CO} = 0.1 \quad (4.6)$$

The weight of the *degree* (D) of an author or a university is:

$$W_D = 0.1 \quad (4.7)$$

The weight of the *closeness centrality* (CL) of an author or a university is:

$$W_{CL} = 0.1 \quad (4.8)$$

The weight of the *betweenness centrality* (B) of an author or a university is:

$$W_B = 0.1 \quad (4.9)$$

A weighted formula for calculating the Academic Capital (AC) of an individual author **I** or a university **U** is shown in Equation 4.10, taking into account various academic performance metrics.

Each term in the equation represents a specific academic performance metric, and the corresponding weight represents the relative importance of that metric in determining the author's overall AC. The equation is as follows:

$$\text{Author}_{AC}(i) = WP(P_i) + WC(C_i) + WHI(HI_i) + WH10(H10_i) + WE(E_i) + WCO(CO_i) + WD(D_i) + WCL(CL_i) + WB(B_i) \dots (4.10)$$

Where i is an Iraqi author.

The Academic Capital of a university, University_{AC} is calculated in Equation 4.11 as follows:

$$\text{University}_{AC}(j) = WP(P_{ij}) + WC(C_{ij}) + WHI(HI_{ij}) + WH10(H10_{ij}) + WE(E_{ij}) + WCO(CO_{ij}) + WD(D_{ij}) + WCL(CL_{ij}) + WB(B_{ij}) \dots (4.11)$$

Where j is an Iraqi university, and **WP (P_{ij})** is the weight for the number of papers (P) published by author i in a university j . **WC (C_{ij})** is the weight for the number of citations (C) received by author i in a university j . **WHI (HI_{ij})** is the weight for the H-Index (HI) of author i in a university j . **WH10 (H10_{ij})** is the weight for the H10-Index (H10) of author i in a university j .

WE (E_{ij}) is the weight for the Eigen Centrality (E) of author i in a university j. **WCO (CO_{ij})** is the weight for the Clustering Coefficient (CO) of author i in a university j. **WD (D_{ij})** is the weight for the Degree Centrality (D) of author i in a university j. **WCL (CL_{ij})** is the weight for the Closeness Centrality (CL) of author i in a university j. **WB (B_{ij})** is the weight for the Betweenness Centrality (B) of author i in a university j.

The weighted formula allows for the customization of the AC calculation, which is based on specific needs or preferences of the organization or individual using it. The weights assigned to each metric reflect the relative importance of that metric in determining the overall AC of the author or university. The weights can be adjusted based on the specific goals or priorities of the organization or individual.

CHAPTER FIVE

Implementation and Results Discussion

5.1 Preface

In this chapter, the result of the proposed system is discussed in detail, the proposed system employs web scraping to collect authors data from Iraqi universities, by utilizing the domains (edu.iq, edu.krd, and uod.ac) using Google Scholar and storing it in a distributed database hosted on the Amazon Cloud. The big data technology is used to generated by Google Scholar profiles is challenging compared to typical data processing methods. The data is too large and complex to be managed and evaluated effectively. Therefor the amount of data generated by academic research is huge, which makes standard data processing and analysis difficult. As it requires complex methods to manage and derive valuable information. Thus, powerful data processing technologies is used in this work such as Apache Spark, parallelism, Virtualization and Cloud Computing to extract useful information from the data. Data is analyzed and visualized to obtain the academic capital. This work aims to propose and present academic capital for Iraqi authors and Iraqi universities by conducting a comprehensive analysis and applying advanced visualization techniques. This attempt not only contributes to a more comprehensive grasp of the scholarly landscape within Iraq; however, it also sheds light on the intellectual contributions that Iraqi authors have made to the academic community all over the world. In addition, it aims to produce useful insights that can contribute to the formation of educational policies, research collaborations, and academic developments.

5.2 Hardware and Software Requirements

In this system, two physical devices are used, namely, Lenovo LEGION laptop, and Lenovo ThinkPad laptops for an environment suitable for experimentation. The details of the two computers are shown in Table 5.1.

Table 5.1 Hardware Requirements

Model name	CPU	RAM	GPU	System Type
Lenovo LEGION	AMD Ryzen 7 5800H with Radeon Graphics 16 CPUs, 3.2GHz	16.00 GB	NVIDIA GeForce RTX 3050Ti Laptop GPU	64-bit
Lenovo ThinkPad	Intel Core i5-4300M 2.6GHz	4.00 GB	VGA HD Intel Graphics	64-bit

Regarding software requirements, a set of important software is needed to show the work experimental nature. The software can be listed by classification into operating systems, programming languages, auxiliary applications, and auxiliary library packages.

- 1) Hardware and Software requirements: In the process of designing, implementing, and experimenting, both Windows and Linux operating systems are required. The hardware used is a Lenovo Legion laptop computer, with the use of several versions of the Linux system. The version installed on the Lenovo Legion computer is 20.04.5 LTS –Focal Fossa-, and the version installed on the Lenovo ThinkPad was 20.04.3 LTS –Focal Fossa- As for version 14.04.2 LTS GNU of Ubuntu Linux, the virtual environment on the virtual box to run scrapers is used.
- 2) Programing language: the system implements Python 3.9.12.(Jupyter Notebook and Integrated Development and Learning Environment (IDLE)). The python language (IDLE) is used in the process of creating scrapers, while Jupyter Notebook is used to process data in the Apatch spark. In addition, helper languages such as JSON is used.
- 3) Auxiliary applications: Auxiliary applications are used in creating the project and during the implementation, experimentation and comparison. Virtual Box is used to obtain a virtual environment that simulates the reality of providing a local network connected to the cloud. In addition to using the Gradle application, Spark projects use Gradle to manage dependencies, compile code, run tests, and generate executable artifacts. The mongo-spark-connector integrates Apache Spark and MongoDB. Gradle manages the mongo-spark-connector build process, including obtaining and resolving dependencies, generating code, running tests, and building the

executable artifact. Gradle simplifies building, testing, and deploying Spark applications that use MongoDB as a data source by automating the mongo-spark-connector library build process.

- 4) PuTTY Configuration: "PuTTY Configuration" refers to the settings and choices of a software application known as "PuTTY." it is a terminal emulator, serial console, and network file transfer tool that is available for free and open source. It is used extensively, and its primary purpose is to set up secure shell (SSH) connections, Telnet sessions, and other network connections utilizing a wide range of protocols.
- 5) Auxiliary libraries: There are many auxiliary packages, including basic and specific ones such as: sys, time, pymongo, pandas, SerpApi, urlsplit, os, csv, re, findspark, pyspark.

5.3 System implementation

Implementation of the system depends on creating a suitable environment with all requirements met. Figure 5.1 represents the outline diagram of the system implementation architecture.

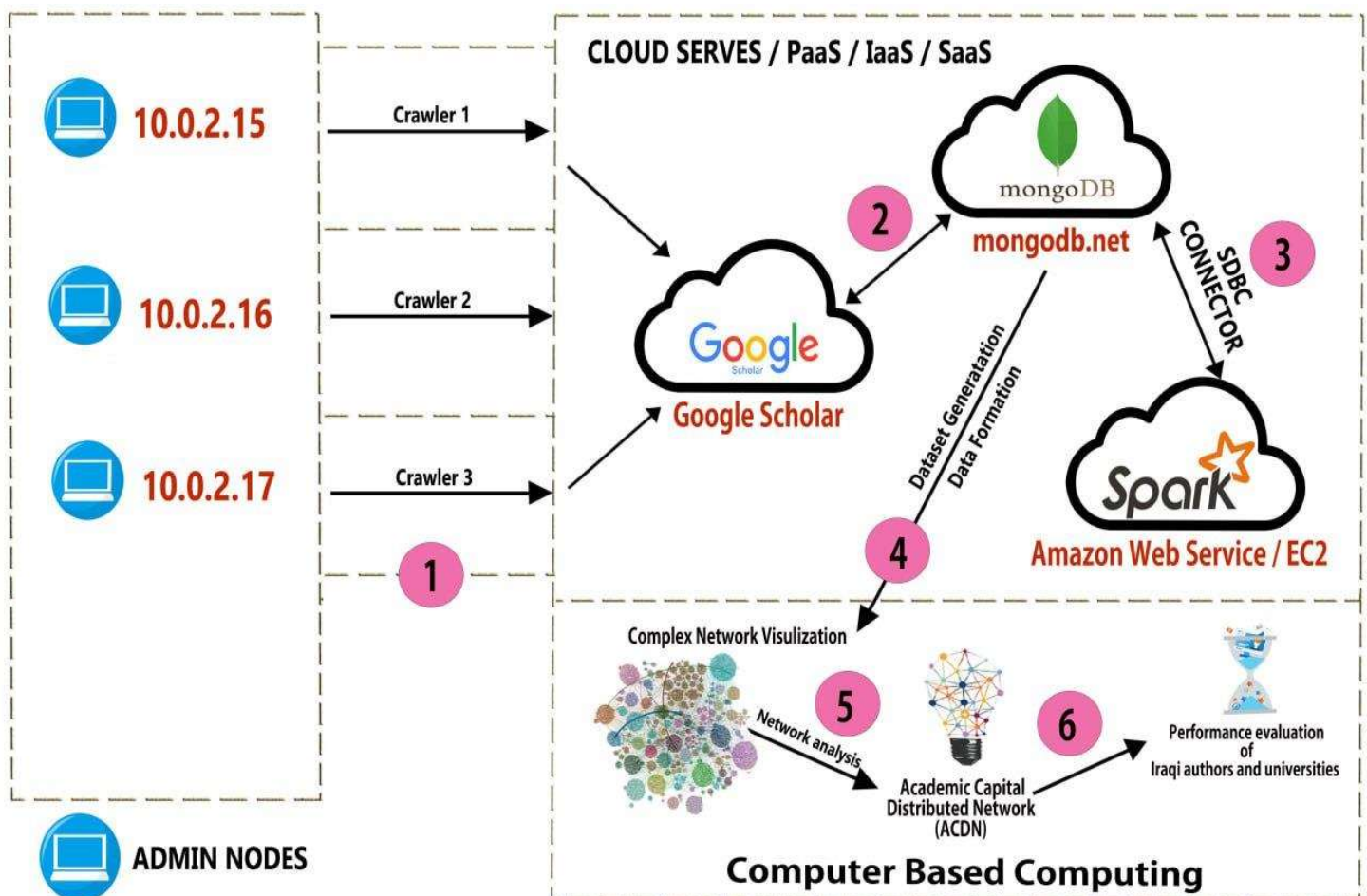


Figure 5.1 System implementation architecture

Figure 5.1 represent system implementation phases, which reflect the way of how the system is operated, via a sequence of numbers that denote the involved phases. Initially, the operation commences with the execution of three different scraper programs, each running on separate computer nodes. These scrapers gather a wealth of data, comprising not only researchers' personal profile files but also their scholarly papers, meticulously extracted from the vast repository of Google Scholar. Once gathered, this dataset securely resides in a Distributed MongoDB Atlas database on the AWS cloud, employing a streaming data format that guarantees real-time accessibility and scalability. This step sets the stage for subsequent data processing and analysis. In the third stage, MongoDB Atlas seamlessly interfaces with Apache Spark, facilitated by the Spark MongoDB Connector. This integration marks the inception of the data processing phase, leveraging the robust capabilities of Apache Spark to demonstrate and manipulate the collected data. The fourth stage begins by generating essential node and edge files, initiating the process of forming and creating the dataset. These files play a vital role in enabling a thorough network analysis, a crucial aspect of the system's functionality. After dataset preparation, the system enters the network visualization stage, powered by the Cytoscape software. Empowered by this component is the capacity to visually explore and interpret complex network structures, thus, enhancing the understanding of the data.

subsequently, in stage five, the analysis phase of the network is initiated, providing insights into the intricacies of network behavior and characteristics. The groundwork for the introduction of a novel ranking metric known as Academic Capital in stage six is laid, which considers multiple criteria to assess the significance and influence of academic entities within the network. The final stage of the process includes a meticulous comparison of data points, enriching the research insights generated by the system.

5.3.1 Data Collection Steps

Web scraping on Google Scholar is used to obtain the data, which includes information of academic publications, citations, authors, and articles, as well as a variety of other details. In order to complete the procedure, it is necessary to access and retrieve data from different web pages on Google Scholar platform. Two models are used in order to perform scraping in an efficiently manner.

For *Web Scraping Parallelism*, three virtual machines (VMs) are operated, with each functioning as a distinct instance of VirtualBox. Each VM is tasked with data retrieval from a specific domain. This embodies a form of parallelism where individual VMs independently operate to extract data from their designated domains. All three VMs collaborate in parallel to concurrently collect data from their respective domains. Each VM functions autonomously and concurrently, focusing on its specific domain. The parallelism here revolves around VMs functioning and retrieving data simultaneously.

5.3.1.1 Unischolar Model

The UniScholar utilizes data collection effectively to offer an overview about the academic environment of Iraqi universities authors. This model creates a smooth connection between university IDs, authors affiliations, and thorough publishing data by combining three scrapers. The following represent the three scrapers implemented in this model:

Scraper1: (Search Organization)

The goal of this Scraper is to retrieve the IDs of all Iraqi universities listed on Google Scholar by executing a web scraper written in Python and utilizing the Scholarly library. The final goal is to collect a dataset with fundamental information about Iraqi universities (name and ID). The name of Iraqi university is used as the first piece of input for the scraper, and it is obtained from the Ministry of Higher Education and Scientific Research in Iraq website, encompassing a total of 126 universities, both government and private. The scraper looks up information based on the provided university's name, so it is able to determine the identifiers of the private and government universities in Iraq. Based on the search results obtained from the scientific researcher at Google scholar, the scraper concentrates on extracting the unique identifiers of each institution.

After implementation, the IDs are collected for all universities that have the identifier on Google Scholar, as shown in Figure 5.2. These ID serve as individual and distinct identifiers for each educational institution.

name	link	id
University of Baghdad	www.en.uobaghdad.edu.iq	2055656215519197438
University of Mosul	www.uomosul.edu.iq/en	10983815109167613928
University of Basrah	www.uobasrah.edu.iq	713597364856705536
Mustansiriyah University	www.uomustansiriyah.edu.iq	00000000000000000000
University of Technology	www.uotechnology.edu.iq	10906564628192137533
University of Kufa	www.uokufa.edu.iq	12593476701188387749
University of Al-Qadisiyah	www.qu.edu.iq	00000000000000000000
Tikrit University	www.tu.edu.iq	6805372774095930672
University of Babylon	www.uobabylon.edu.iq	1999031006410087235
University of Kerbala	www.uokerbala.edu.iq	00000000000000000000
University of Thi-Qar	www.utq.edu.iq	00000000000000000000
Misan University	www.uomisan.edu.iq	00000000000000000000
Al-Nahrain University	www.nahrainuniv.edu.iq/ar	00000000000000000000
University of Diyala	www.uodiyala.edu.iq	18342936164438069611
University of Anbar	www.uoanbar.edu.iq	18567763899000544123
Wasit University	www.uowasit.edu.iq	00000000000000000000
University of kirkuk	www.uokirkuk.edu.iq	00000000000000000000

Figure 5.2 Sample of Identifiers for the Iraqi Universities

It should be mentioned, however, that implementation shows a limitation on scraping process which must be taken into consideration. The results reveal that only *nine* universities have IDs available on Google Scholar. Despite this, the scraper is still beneficial as it successfully collects useful data for educational institutions that have university IDs. The nine universities with IDs as listed below:

University of Mosul, University of Baghdad, University of Basrah, University of Technology, University of Kufa, Tikrit University, University of Babylon, University of Diyala, University of Anbar.

Scraper2: (Search Author by organization)

This scraper has a specific objective, which is collecting data represented by researchers' affiliation from the nine universities listed above in scraper1. The university ID, which was received from the first scraper, represent input to scraper2 which focuses on Google Scholar searches to obtain information about all researchers connected to universities. A lot of data is gathered a lot of data, including the essential information for each author as shown in Figure 5.3 and listed below:

- ❖ *Authors' Names:* The scraper retrieves full names of all of the researchers' who are affiliated with the given universities. This confirms that the researchers can be correctly and completely identified.
- ❖ *Affiliation:* The scraper obtains each researcher's institutional affiliation, which identifies the precise university to which they are associated with.

Understanding the academic network and institutional ties of the researchers depends heavily on this information.

- ❖ *Author Profile Image*: The scraper is made to retrieve the researchers' profiles images whenever they are available.
- ❖ *Cited By*: This data point shows how frequently a researcher's work has been referenced by other academics. It offers detail about the effect and sway that the researcher's writings have had on the academic world.
- ❖ *Interests*: Information on each researcher's areas of expertise and research interests is gathered by the scraper. Finding potential collaborators with a similar research emphasis and forming academic alliances both depend on an understanding of the scholars' interests.
- ❖ *Author ID*: Each researcher is given a special identification number, or author ID, as part of the data collection procedure by the scraper.

```
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'ns3IPiIAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=ns3IPiIAAAA', 'name': 'Dena Al-Abdulla', 'affiliation': 'university of mosul', 'email_domain': '@uomosul.edu.iq', 'interests': ['inorganig chemistry'], 'citedby': 7933}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': '6Fov68SAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=6Fov68SAAAA', 'name': 'MA Alsalem', 'affiliation': 'University of Mosul, College of Administration and Economics', 'email_domain': '@uomosul.edu.iq', 'interests': ['Decision Theory', 'Managment Information System', 'Tele-medicine'], 'citedby': 3386}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'mFg4hnMAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=mFg4hnMAAAA', 'name': 'Yasser Fakri Mustafa', 'affiliation': 'Professor in Pharmaceutical Chemistry, College of Pharmacy, University of Mosul', 'email_domain': '@uomosul.edu.iq', 'interests': ['Pharmacy', 'Pharmaceutical chemistry', 'Chemistry of natural products'], 'citedby': 3002}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'Je6QRTUAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=Je6QRTUAAAA', 'name': 'Zakariya V. Algama', 'affiliation': 'University of Mosul, College of Computer Science and Mathematics', 'email_domain': '@uomosul.edu.iq', 'interests': ['high dimensional data', 'penalized (regularized) methods and', 'gene selection in cancer classification', 'Bioinformatics', 'QSAR (chem)', 'citedby': 2409}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'ebK5frcAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=ebK5frcAAAA', 'name': 'Abdelrahman B. Fadhill', 'affiliation': 'University of Mosul College of Science', 'email_domain': '@uomosul.edu.iq', 'interests': ['Chemistry'], 'citedby': 2058}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'v2DR300AAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=v2DR300AAAA', 'name': 'Muhsin W. Mohammed', 'affiliation': '(Ph.D) in physics, Department of Physics, College of Education for Pure Science, University ...', 'email_domain': '@uomosul.edu.iq', 'interests': ['Nuclear Physics', 'Experimental Physics', 'Elementary Particle Physics'], 'citedby': 1418}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'xRarbvCAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=xRarbvCAAAA', 'name': 'LBY Aldabbagh', 'affiliation': 'University Of Mosul, College of Engineering, Professor of mechanical Engineering', 'email_domain': '@uomosul.edu.iq', 'interests': ['Energy', 'CFD', 'Porous Media', 'Domestic Water heater'], 'citedby': 1348}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'VIP-PJMAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=VIP-PJMAAAA', 'name': 'Dr. Mohammed Hazim Alkawaz', 'affiliation': 'Department of Computer Science , College of Education for Pure Science , University of ...', 'email_domain': '@uomosul.edu.iq', 'interests': ['Multimedia', 'Cybersecurity', 'Image processing', 'Facial Animation', 'Virtual Human'], 'citedby': 1297}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'zgC1A4UAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=zgC1A4UAAAA', 'name': 'Fouad K. Mohammad', 'affiliation': 'University of Mosul, College of Veterinary Medicine, Professor Emeritus, Pharmacology and ...', 'email_domain': '@uomosul.edu.iq', 'interests': ['Toxicology', 'Pharmacology', 'Neurotoxicology', 'Cholinesterase', 'Organophosphates'], 'citedby': 1212}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'SnLIZkKAAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=SnLIZkKAAAA', 'name': 'Abdul-Rahim Ibrahim Al-Hadidy', 'affiliation': 'University of Mosul, College of Engineering', 'email_domain': '@uomosul.edu.iq', 'interests': ['Paving materials', 'design', 'polymers', 'sustainability', 'recycling materials'], 'citedby': 1159}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'RurC0d4AAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=RurC0d4AAAA', 'name': 'salah YOUSIF', 'affiliation': 'University of Mosul - College of Pharmacy', 'email_domain': '@uomosul.edu.iq', 'interests': ['pharmacology', 'clinical pharmacy', 'addictology'], 'citedby': 1013}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'xxakRh0AAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=xxakRh0AAAA', 'name': 'Nasir Muwfaq Younis', 'affiliation': 'University of Mosul, College of Nursing', 'email_domain': '@uomosul.edu.iq', 'interests': ['Community Health Nursing'], 'citedby': 929}
{'container_type': 'Author', 'filled': [], 'source': <AuthorSource.SEARCH_AUTHOR_SNIPPETS: 'SEARCH_AUTHOR_SNIPPETS'>, 'scholar_id': 'eqILM48AAAA', 'url_picture': 'https://scholar.google.com/citations?view_op=medium_photo&user=eqILM48AAAA', 'name': 'AbdulRahman Hani Aldaood', 'affiliation': 'University of Mosul, College of Engineering, Civil
```

Figure 5.3 Sample Data from a Specific University

The scraper is able to efficiently classify and store the data connected with each researcher and their institution affiliation after receiving these identifiers and using them in the process.

This model collects data from only (9) universities in Iraq out of (126) Iraqi universities. Thus, solved this issue of data collecting in order to include all Iraqi universities (private and government) a number of different steps are considered. Firstly, Google support teams was asked for assistance in obtaining university IDs, and more than one responsible person from Google support team was contacted for assistance. Despite multiple attempts of contacting the

support, there has been no solution response to this particular issue. Additionally, proactive steps were taken, which included communicating with the presidency of the University of Mosul, which then approached the Ministry of Higher Education and Scientific Research. These contacts intended to seek advice and guidance in resolving the obstacles of gaining access to crucial university identifiers. In spite of all of these efforts, this approach has not yielded a workable solution.

As a result, an alternative approach was proposed to solve this limitation by implementing a new method which will explained in the second proposed model (ScholarSync model). The proposed model describes methodical procedure in order to overcome research gap and complete the work successfully.

5.3.1.2 ScholarSync Model

This program interacts with Google Scholar in a smooth manner, obtaining detailed data about researchers, their profiles, publications, and academic accomplishments. The scraper provides rapid and precise retrieval of authors' data by employing SerpApi, offering a useful resource for research analysis. Streamlining data collection is achievable with the assistance of this scraper, allowing a focus on extracting pertinent conclusions from the extensive data available on Google Scholar. After collecting the data, it is stored in the Cloud MongoDB Atlas, a cloud-based distributed database management system. MongoDB Atlas was chosen as it provides high scalability, flexibility, and availability, which are crucial for managing large amounts of data. The data of the authors collected from Google Scholar using ScholarSync model for Iraqi universities amounted to 44,290 authors, and the data of the published papers of these authors amounted to 528,023 research papers. This model was implemented by suggesting the following scrapers:

Scraper1: (Google-Scholar-Profiles)

Three Python programs were written to be executed concurrently on the three virtual machines in order to get more data in a minimum amount of time. These programs were written as follows:

- The first program scrapes all Iraqi researchers' data within the domain *edu.iq*

[illegible]

Scraper2: (Google-Scholar-Author)

ScholarSync Model strategy for web scraping retrieves a large amount of information efficiently, which gives significant insights into the researcher's academic contributions and successes. The obtained data from all authors, that

is, equal 44290 authors in Iraqi universities consists of: *thumbnail, name, affiliations, email, website, interests, cited-by-table, cited-by-graph, public-access-link, available-public-access, co-authors, not-available-public-access*.

research papers collected from each author, that is, 528,023 research papers consists of: *name of author, affiliations, email domain, image, interest, article-title, article-link, article-citation-id, article-authors, article-publication, article-cited-by-value, article-cited-by-link, article-cited-by-cites-id*. All data in regards to authors and research paper are as shown in Figure 5.6.

ScholarSync model provides efficient retrieval of large amount of information about authors, and its implementation is based on the following steps:

1. Domains are partitioned into three subgroups using a list of names associated with Iraqi domains. Each subset corresponds to a section of the Iraqi researcher's data.
2. Three different VM was used to implement parallel processing, this produces three parallel scraper instances, each of which is responsible of gathering information from a different section of the Iraqi domain.
3. Each scraper instance sends HTTP queries to Google Scholar with the proper search criteria or query strings in order to retrieve the information for the relevant subset of the Iraqi domain.
4. parser library was used to process the HTML answers that Google Scholar sends and extract the required information. the appropriate HTML elements, classes, or patterns that contain the necessary data, such as author names, publication titles and et al, are recognized.
5. The extracted data was stored in a manner that can handle data conflicts or synchronization problems in CSV file of the Cloud.
6. To handle unsuccessful queries and other possible scraping-related issues, retries and error handling were employed using Api. It's critical to manage exceptions and make sure the scraping process can go on without interruption.
7. Once all parallel operations were finished, the extracted data from each was merged and consolidated into a single dataset.

A1	thumbnail,name,affiliations,email,website,interests,cited_by_table,cited_by_graph,public_access_link,available_public_access,not_available_public_access,co_authors
A	
1	thumbnail,name,affiliations,email,website,interests,cited_by_table,cited_by_graph,public_access_link,available_public_access,not_available_public_access,co_authors
2	https://scholar.googleusercontent.com/citations?view_op=view_photo&user=flBwOEAAAAJ&citpid=7,Karzan Abdulmuhsin Mohammad,"Department of Biology, College of Education, Salahaddin University-Erbil",Verified email at su.edu.krd,https://academ
3	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=zR1huCcAAAAJ&citpid=13,Dara K. Mohammad,Salahaddin University-Erbil,Verified email at su.edu.krd,,"[title: 'Cancer Cell Signaling', link: 'https://scholar.google.com/citatio
4	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=mEIDJYAAAAJ&citpid=4,Omid Amiri,"Department of Chemistry, University of Raparin, Kurdistan Region, Iraq",Verified email at uor.edu.krd,http://uor.edu.krd/nrg/index.php?e
5	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=mee4yLYAAAAJ&citpid=8,Ali H. A. Al-Waeli,"Engineering Department, American University of Iraq - Sulaimani, Kurdistan Region, Iraq",Verified email at aus.edu.krd,https://au
6	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=FLnXzdcAAAAJ&citpid=13,Suhbi R. M. Zeebaree,"Prof. Dr. Eng. of Computer Engineering, Duhok Polytechnic University",Verified email at dpu.edu.krd,https://staffportal.dpu.e
7	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=awOOXwAAAAJ&citpid=4,Pawan Faris,"Department of Biology, College of Science, Salahaddin University-Erbil",Verified email at su.edu.krd,http://su.edu.krd/,"[title: 'Molecul
8	0,https://scholar.google.com/citations/images/avatar_scholar_128.png,Muhsen Yaseen, School of Business,Verified email at ukh.edu.krd,,"[citations: {all: 3164, 'since_2017': 2230}], {h_index: {all: 1, 'since_2017': 1}], {i10_index: {all: 1, 'since_2017':
9	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=DpyjzKMAAAAAJ&citpid=1,Mohammad Raziq Al Imami,"The American University of Iraq, Sulaimani (AUIS)",Verified email at aus.edu.krd,https://aus.edu.krd/faculty-listing,"[title
10	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=5K0ZdwEAAAAJ&citpid=3,Dr. Govand Anwar,Global University of Erbil,Verified email at gue.edu.krd,,"[title: 'Business Administration', link: 'https://scholar.google.com/citations
11	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=PIPepUAAAAJ&citpid=9,Tarik A. Rashid,"Computer Science and Engineering, University of Kurdistan Hewler, Iraq",Verified email at ukh.edu.krd,https://scholar.google.com/cita
12	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=e5McLU8AAAAJ&citpid=1,Varoujan Sissakian,"Natural Resource Eng. & Mang., University of Kurdistan Hewler, Kurdistan Region, Iraq",Verified email at ukh.edu.krd,https://ww
13	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=aBdgHxAAAAJ&citpid=2,Prof. Dr. Eng. Adnan Mohsin Abdulazeez, Duhok Polytechnic University,Verified email at dpu.edu.krd,,"[title: 'Soft Computing and Machine Learning'
14	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=LOLeKNOAAAAJ&citpid=1,Kayhan R. Z. Ghafour,"Department of Software Eng., Salahaddin University-Erbil, Iraq",Verified email at su.edu.krd,,"[title: 'Vehicle-to-everything (V
15	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=cJ2PraIAAAAAJ&citpid=6,Bahram Resul,Associate professor,Verified email at ukh.edu.krd,,"[title: 'Medicine', link: 'https://scholar.google.com/citations?view_op=search_autho
16	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=37r708AAAAJ&citpid=5,Amanj Baker,"College of Pharmacy, Hawler Medical University, Kurdistan Region- Iraq",Verified email at hmu.edu.krd,https://hmu.edu.krd/profile?ectu
17	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=uuuBYtAAAAJ&citpid=2,Nawfal R Hussein,"Department of Medicine, College of Medicine, University of Zakho, Kurdistan Region, Iraq",Verified email at uoz.edu.krd,http://staf
18	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=RtfH3gwAAAAJ&citpid=1,Omer Hasan Ali,Unknown affiliation,Verified email at kti.edu.iq,,"[title: 'omer.ali@kti.edu.iq', link: 'https://scholar.google.com/citations?view_op=searc
19	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=Hkq7WwAAAAJ&citpid=5,ismail hamadameen sleman,"Kurdish department, college Education of Shaqlawa, Sallahadin University",Verified email at su.edu.krd,,"[title: 'literat
20	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=E_jcC0sAAAAJ&citpid=1,Riadh Francis,Assistant Professor of Microbiology,Verified email at cue.edu.krd,,"[title: 'General Microbiology', link: 'https://scholar.google.com/citati
21	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=jn01wEAAAAJ&citpid=53,Shuokr Qarani Aziz,"Department of Civil Engineering, College of Engineering, Salahaddin University-Erbil",Verified email at su.edu.krd,https://academ
22	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=tu902SAAAAJ&citpid=1,Robert Manning,American University of Kurdistan,Verified email at auk.edu.krd,http://www.creditcardnation.com,"[title: 'Immigration', link: 'https://sc
23	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=y4dsm-kAAAAJ&citpid=2,Mohammed A. M. Sadeeq, PhD Student, Duhok Polytechnic University",Verified email at dpu.edu.krd,https://staffportal.dpu.edu.krd/mohammed,"[title
24	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=zWiyylAAAAJ&citpid=17,soad najmaddin mohalidin khaffaf,"Department of Chemistry, College of education,Salahaddin University-Erbil",Verified email at su.edu.krd,https://aca
25	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=5H_1QBAAAAJ&citpid=3,Rizgar R. Zebari,Nawroz University,Verified email at nawroz.edu.krd,,"[title: 'Networking', link: 'https://scholar.google.com/citations?view_op=search_a
26	0,https://scholar.googleusercontent.com/citations?view_op=view_photo&user=m23r-iOAAAAJ&citpid=2,Muhammed Abdulbaki Ibrahim,"Professor of Electronics Engineering, Electrical Eng. Dept., Salahaddin University-Erbil",Verified email at su.edu.krd,ht

A1	,name,article_title,article_link,article_year,article_citation_id,article_authors,article_publication,article_cited_by_value,article_cited_by_link,article_cited_by_cites_id
A	
72	70,Karzan Abdulmuhsin Mohammad,"The global burden of cancer attributable to risk factors, 2010–19: a systematic analysis for the Global Burden of Disease Study 2019",https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAA
73	71,Karzan Abdulmuhsin Mohammad,"Erratum: Estimates of global, regional, and national incidence, prevalence, and mortality of HIV, 1980-2015: The Global Burden of Disease Study 2015 (The Lancet HIV (2016) 3 – –https://scholar.google.com/citations?
74	72,Karzan Abdulmuhsin Mohammad,"Erratum: Estimates of global, regional, and national incidence, prevalence, and mortality of HIV, 1980-2015: The Global Burden of Disease Study 2015 (The Lancet HIV (2016) 3 – –https://scholar.google.com/citations?
75	73,Karzan Abdulmuhsin Mohammad,"Global, regional, and national burden of colorectal cancer and its risk factors, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019",https://scholar.google.com/citations?view_op=view_citatio
76	74,Karzan Abdulmuhsin Mohammad,Epidemiological Approach of SARS-CoV2 in the First Month of Appearance in the Kurdistan Region of Iraq,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio
77	75,Karzan Abdulmuhsin Mohammad,Determination the site of antibiotic resistance genes in Escherichia coli isolated From Urinary Tract Infection,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio
78	76,Karzan Abdulmuhsin Mohammad,"Global, regional, national, and selected subnational levels of stillbirths, neonatal, infant, and under-5 mortality, 1980–2015: a systematic analysis for the Global Burden of –https://scholar.google.com/citations?view_o
79	77,Karzan Abdulmuhsin Mohammad,"The role of oxidative stress and haematological parameters in relapsing-remitting multiple sclerosis in Kurdish population,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio
80	78,Karzan Abdulmuhsin Mohammad,"The status of cancer publications in the Kurdistan region of Iraq,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio_for_view=flBwOEAAAAJ:ISLTnruKqC,2
81	79,Karzan Abdulmuhsin Mohammad,"Erratum: Global, regional, national, and selected subnational levels of stillbirths, neonatal, infant, and under-5 mortality, 1980–2015: a systematic analysis for the Global –https://scholar.google.com/citations?view_op
82	80,Karzan Abdulmuhsin Mohammad,"The global burden of adolescent and young adult cancer in 2019: a systematic analysis for the Global Burden of Disease Study 2019,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAA
83	81,Karzan Abdulmuhsin Mohammad,"Global, regional, and national levels of neonatal, infant, and under-5 mortality during 1990-2013: a systematic analysis for the Global Burden of Disease Study 2013",https://scholar.google.com/citations?view_op=view_cit
84	82,Karzan Abdulmuhsin Mohammad,"Cancer Incidence, Mortality, Years of Life Lost, Years Lived With Disability, and Disability-Adjusted Life Years for 29 Cancer Groups From 2010 to 2019: A Systematic Analysis –https://scholar.google.com/citations?view_op
85	83,Karzan Abdulmuhsin Mohammad,"Policy and technical considerations for implementing a risk-based approach to international travel in the context of COVID-19,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pag
86	84,Karzan Abdulmuhsin Mohammad,Association study of Interleukin 10 gene polymorphisms in Iraqi patients with multiple sclerosis,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio_for_view=fl
87	85,Karzan Abdulmuhsin Mohammad,"Correction to Lancet Gastroenterol Hepatol 2020
88	86,Karzan Abdulmuhsin Mohammad,Knocking out the pptA gene in Aspergillus fumigatus could result in new protein profile and promising targets for antifungals,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pages
89	87,Karzan Abdulmuhsin Mohammad,Drinking to death: the changing face of liver disease,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio_for_view=flBwOEAAAAJ:ZHo1McVdvXMC,2018,flBw
90	88,Karzan Abdulmuhsin Mohammad,"Alcohol use and burden for 195 countries and territories, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016",https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBw
91	89,Karzan Abdulmuhsin Mohammad,Measuring progress from 1990 to 2017 and projecting attainment to 2030 of the health-related Sustainable Development Goals for 195 countries and territories: a systematic –https://scholar.google.com/citations?view_o
92	90,Karzan Abdulmuhsin Mohammad,"Erratum: Population and fertility by age and sex for 195 countries and territories, 1950–2017: a systematic analysis for the Global Burden of Disease Study 2017 (The Lancet –https://scholar.google.com/citations?view
93	91,Karzan Abdulmuhsin Mohammad,"Erratum: Global, regional, and national age-sex-specific mortality and life expectancy, 1950–2017: a systematic analysis for the Global Burden of Disease Study 2017 (The Lancet –https://scholar.google.com/citatio
94	92,Karzan Abdulmuhsin Mohammad,"Smoking prevalence and attributable disease burden in 195 countries and territories, 1990-2015: a systematic analysis from the Global Burden of Disease Study 2015 (vol 389, pg –https://scholar.google.com/citation
95	93,Karzan Abdulmuhsin Mohammad,"Erratum: Global, regional, and national levels of maternal mortality, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015 (The Lancet (2016) 388 (10053 –https://scholar.google.com/citatio
96	94,Karzan Abdulmuhsin Mohammad,"Erratum: Estimates of global, regional, and national incidence, prevalence, and mortality of HIV, 1980-2015: The Global Burden of Disease Study 2015 (The Lancet HIV (2016) 3 –https://scholar.google.com/citations?
97	95,Karzan Abdulmuhsin Mohammad,Erratum: Correction to Lancet HIV (2016) 3 (e361–e367)(S253230181630087X)(10.1016/S2352-3018 (16) 30087-X),https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio
98	96,Karzan Abdulmuhsin Mohammad,"Global, regional, and national levels of maternal mortality, 1990-2015: a systematic analysis for the Global Burden of Disease Study 2015",https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBw
99	97,Karzan Abdulmuhsin Mohammad,Transcriptome analysis of quorum sensing and gene interactions of Candida albicans and Pseudomonas aeruginosa in mixed biofilms with regard to the antifungal resistance,https://scholar.google.com/citations?view_op=
100	98,Karzan Abdulmuhsin Mohammad,Multiple Sclerosis and Related Disorders,https://scholar.google.com/citations?view_op=view_citation&hl=en&user=flBwOEAAAAJ&pagesize=100&citatio_for_view=flBwOEAAAAJ:D03IK_w7-QYC,flBwOEAAAAJ:D03IK

Figure 5.6 Retrieval Researcher's Information by Iraqi Domain

5.3.1.3 Overcoming Challenges in Scholarly Data Scraping

After using the suggested scrapers of ScholarSync model, many *problems* have been solved during the scraping process:

1. *CAPTCHA Challenges*: a request was sent to Google Scholar to access their API to have the ability to perform site scraping. Researchers and developers can rapidly and structurally retrieve scholarly data with the help of Google Scholar's API. Captcha Challenges was handled by the private key obtained after submitting request to the SerpApi team and Google Scholar services. The key was obtained after 4 months of starting the work. This key helps to overcome IP blocking or rate limiting problems imposed by search engines (Google Scholar) by rotating a set of proxies and automatically handling IP rotation.
2. There was support from the SerpApi team represented by 60,000 free searches without incurring any cost as shown in Figure 5.7.

An article detailing the experience and specific project on collecting data from Google Scholar using the proposed scraper was composed and published on the "Medium platform." This widely used online publishing platform allows users to create and share articles. Alongside the article, a directive was received from SerpApi team to publish the code related to the web scraping project on a GitHub Repository. GitHub, which is a platform for hosting and sharing code repositories, now hosts the code for the project, and is available at the following link:

<https://github.com/naghamajeel/scrape-from-GS>.

The published article can be accessed at the provided link.

<https://medium.com/@naghamajeel/how-to-scrape-google-scholar-data-using-cloud-computing-serpapi-6e8e7a0451de>.

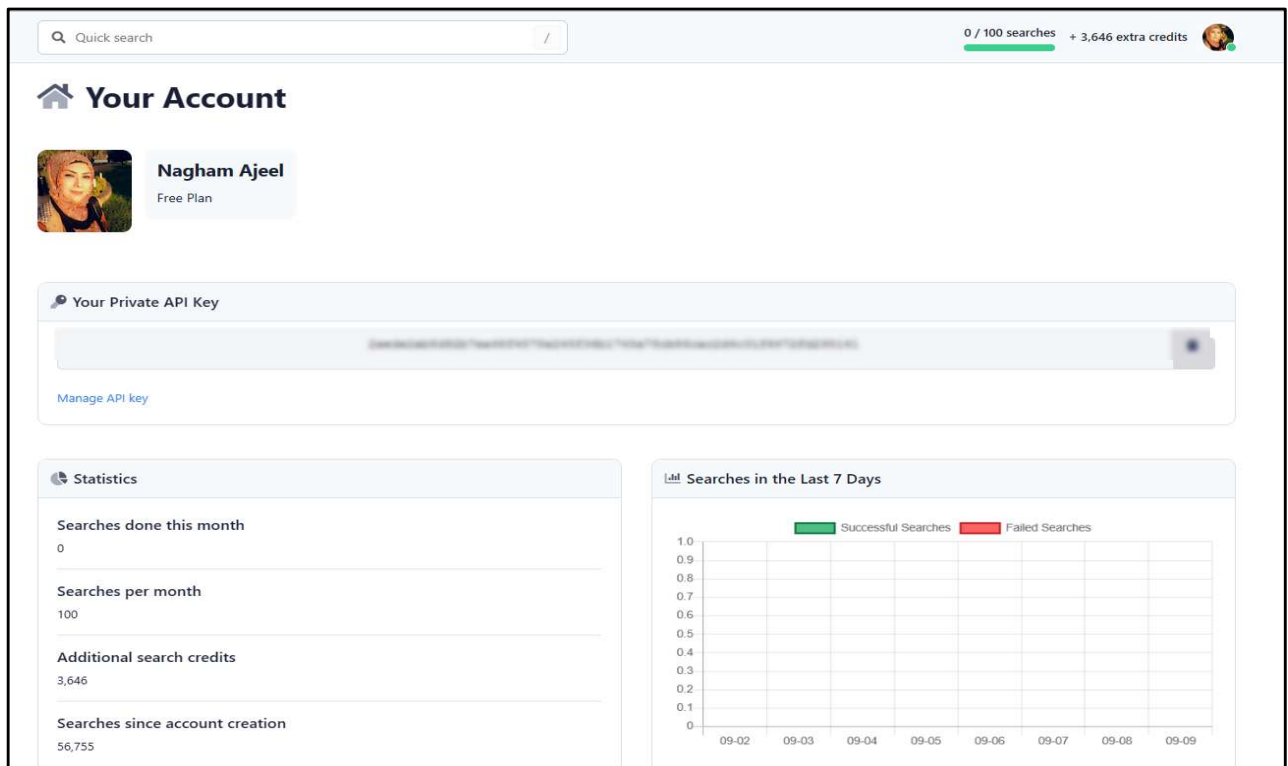


Figure 5.7 Support from SerpApi Team with 60,000 Searches

3. One of the frequent difficulties encountered during the web scraping process was the necessity to scroll through multiple pages of search results. When the required information is distributed across various pages, gathering all the data in a single request becomes challenging. The issue of turning pages (pagination), is responsible for prolonging the scraping process and requiring additional effort to extract the entirety of the data. In web scraping, a challenge emerges when it is not possible to obtain all the necessary data with a single request to a website.

So, pagination, which denotes the practice of splitting content across multiple pages of numbered or next/previous page indicators, is commonly termed the issue of turning pages, due to the requirement for the scraper to navigate through these pages systematically. Encountering pagination during web scraping can have a notable impact on the scraping process in various ways, including:

- ❖ **Time-Consuming:** Scraping multiple pages one by one can be time-consuming, especially if there are a large number of pages to go through.

- ❖ Complexity: Handling pagination often requires more complex code and logic in web scraper to follow the page links, extract data, and store it properly.
- ❖ The necessity for additional effort arises when dealing with pagination in web scraping. This involves iterating through multiple pages, extracting data from each page, and ensuring that no relevant information is overlooked.

4. During implementation and experiments, an error was encountered where in characters from outside the Basic Multilingual Plane (BMP) were present in the data. Characters such as emojis, falling outside the BMP, posed a challenge as web mining was unable to handle these characters. This issue was resolved by employing Unicode that extends beyond the basic multilingual plane.

5.3.2 Managing and Storing Raw Data

The data collected was placed in a Cloud MongoDB Atlas Cluster that was of the replica type as shown in figure 5.8, to guarantee a safe and dependable method of data storage. By copying the dataset and distributing it over several nodes within the cluster, the replica cluster setup insured that there would be high availability of the data as well as data redundancy. A configuration of a three-node in a cluster was established, as shown in Figure 5.9, where three independent instances of MongoDB were executed within separate containers. This arrangement ensures both high availability and data redundancy. A replica set was formed by a group of database servers that collaborate in MongoDB to maintain multiple copies of identical data. This setup provides redundancy and fault tolerance, enabling automatic failover. The primary node handles all write operations, while the secondary nodes replicate the data and can take the role of the primary node if necessary.

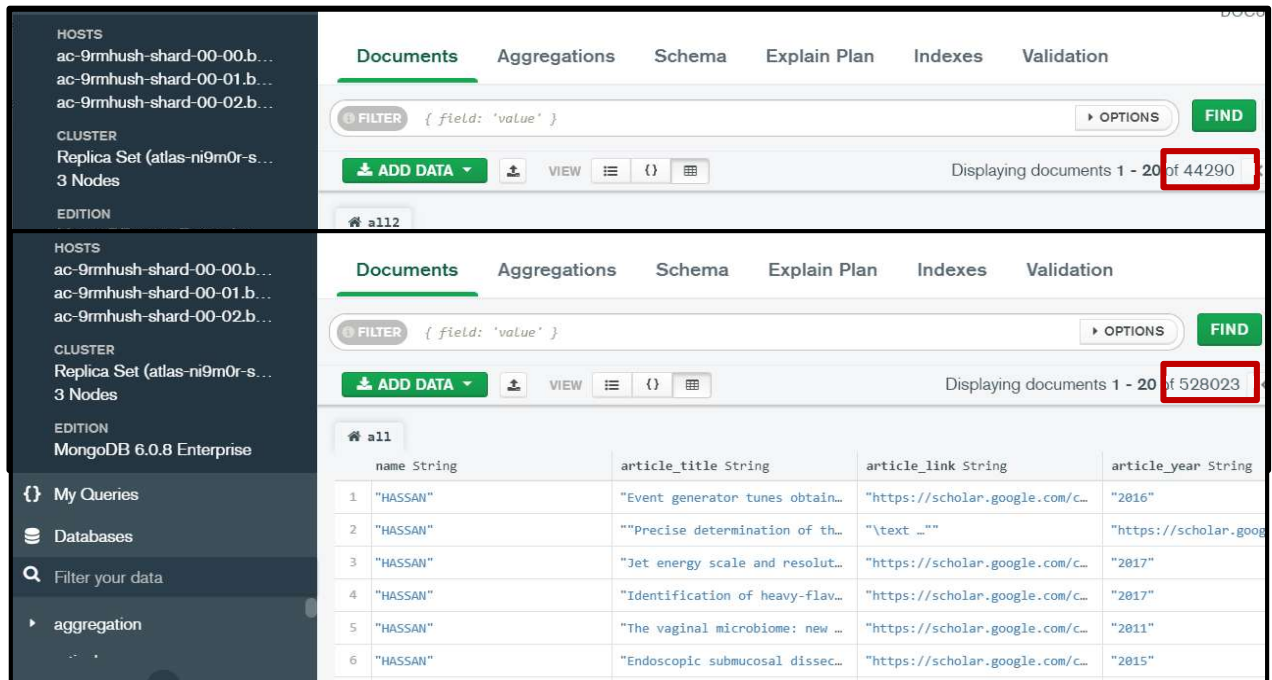


Figure 5.8 Store Streaming Data in Cloud

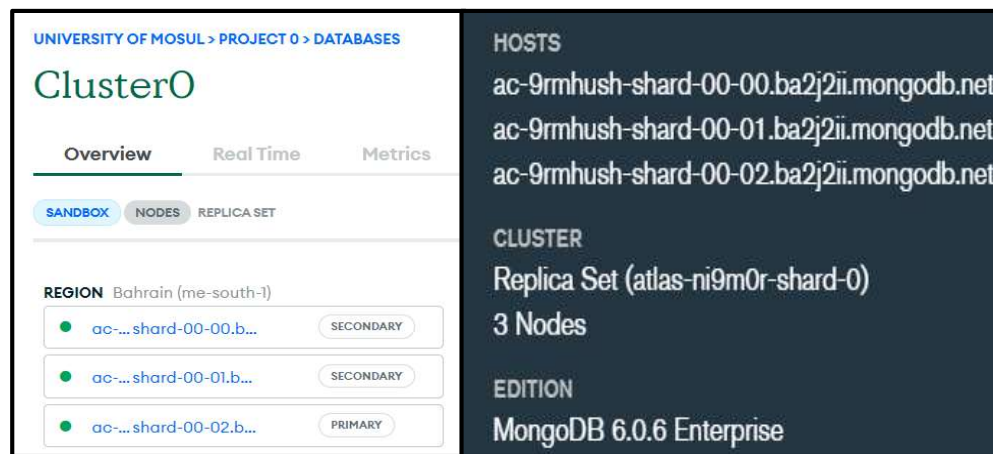


Figure 5.9 Configuration of Replica Cluster

Several significant indicators related to data transfer and connection inside the MongoDB cluster were included in MongoDB Atlas. This cluster's network activity, connectivity, and data storage features can be better understood by examining the metrics for the "network," "connection," and "logical size" respectively, as shown in Figure 5.10, 5.11 and 5.12. These parameters were monitored and analyzed to improve cluster performance, guarantee effective data transfer, and make wise scaling decisions.



Figure 5.10 Network Indicators in Cluster



Figure 5.11 Logical Size Indicators in Cluster



Figure 5.12 Connections Indicators in

In the MongoDB Atlas cloud environment, a cluster was configured with three nodes of the replica set type to store data with parallelism. The core advantage lies in the ability to parallelize data storage operations, providing concurrent read and write capabilities for improved throughput. The replica set comprises a primary node responsible for handling write operations, and two secondary nodes that replicate data from the primary. Fault tolerance and high availability are ensured by this architecture. In the event of a node failure, the system seamlessly switches to the available nodes, maintaining continuous access to data. Parallelism was incorporated into data storage through a replica set, allowing simultaneous writing and reading from privileged nodes. This concurrent operation enhances both read and write throughput, ensuring fault tolerance.

It is worth noting that all nodes, including primary and secondary, in the cluster were monitored by the “network”, “connection”, and “logical size” parameters, as in Figure 5.13. Network metrics in MongoDB focus on the communication between the database server and clients. *Network metrics*, such as "BytesIn" and "BytesOut," shed light on data transfer between nodes and clients, aiding in effective network bandwidth management.

In MongoDB, *connection metrics* revolve around monitoring the number of active connections to the database. The "current" indicator in the connection metrics refers to the current number of active connections to the MongoDB cluster at a given point in time. This metric provides real-time information about the number of client connections actively interacting with the database. It plays a key role in performance monitoring, and identifying potential issues related to the demand on the MongoDB cluster.

The *Logical Size* metric measures the cumulative size of data at a given point in time within a MongoDB database, providing an indication of the space occupied by documents and indexes in a logical sense. In summary, a scalable, and high-performance environment in the cloud is crafted by the orchestrated MongoDB Atlas cluster, featuring a three-node replica set and parallel storage capabilities. The strategic use of parallelism, coupled with monitoring of key metrics, ensures a responsive system capable of handling varying workloads and demands.



Figure 5.13 Parameters for the Primary and Secondary Node

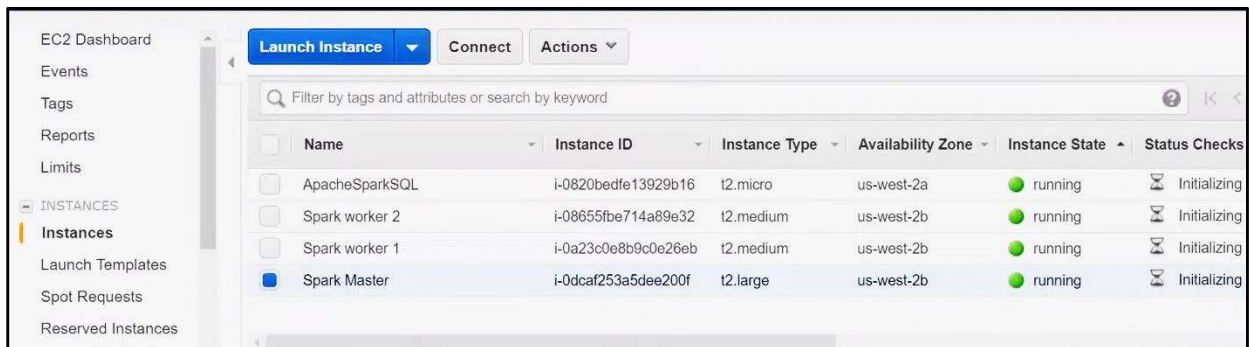
5.3.3 Parallel Data Processing

The processing environment in the Cloud was commenced, by encompassing a JupyterLab instance, a MongoDB cluster, and Apache Spark deployment of with one master node and three worker nodes. With the deployment of Spark on the AWS cluster, the data was analyzed and manipulated in a distributed manner. Spark's parallel processing abilities allow for the execution of complex tasks on large datasets by dividing them into smaller partitions and processing them simultaneously across the cluster's nodes. This parallelization significantly was enhancing the performance and speed of data processing operations. The aggregation of data from various sources has resulted in big datasets. In the context of Iraqi authors data, the quantity of information obtained from Google Scholar exceeds the

computational capabilities of traditional single-node systems. Because as the size of the data set increases, issues with data storage, processing time, and memory limits increase as well. PuTTY, which is a popular SSH client, was used to securely access a Spark cluster running on cloud EC2 instance in Amazon Web Services.

The cluster that has been created comprises virtual servers, referred to as instances, composed of four nodes: one master and three worker nodes, each deployed on virtual servers. To acquire these devices, Amazon's ec2 instance was utilized, with the Master Node of type *t2.large*, featuring 2 CPUs and 8GB of RAM, and three workers of type *t2.medium*, equipped with 2 CPUs and 4GB of RAM each. After creating all devices in the AWS cloud, initiation of the master device was achieved by configuring PuTTY and utilizing the public DNS (ip4). When the cloud is successfully logged in, the Spark Cluster can be accessed, which eventually leads to access to the command line of the EC2 instance. In order to access an EC2 instance, the instance in question needs to be located in a public subnet and have suitable security group rules that permit SSH traffic to originate from the user's IP address.

Also, the Worker runs by entering the public DNS (ip4) of each of their nodes and calling up the private key, in this case, all instances run, each in a private window. Figure 5.14 Shows Running of all nodes in the cluster.



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
☐	ApacheSparkSQL	i-0820bedfe13929b16	t2.micro	us-west-2a	running	Initializing
☐	Spark worker 2	i-08655fbe714a89e32	t2.medium	us-west-2b	running	Initializing
☐	Spark worker 1	i-0a23c0e8b9c0e26eb	t2.medium	us-west-2b	running	Initializing
☒	Spark Master	i-0dcdf253a5dee200f	t2.large	us-west-2b	running	Initializing

Figure 5.14 Running Spark Cluster

After the Spark Cluster was configured, Spark was executed on each node, as illustrated in Figure 5.15. This is the reason why a distributed system was implemented, providing a compelling solution to address the constraints associated with conventional processing techniques.


```

ec2-user@ip-172-31-9-130:~/sparkk
┌───┐ ┌───┐ ┌───┐
└───┘ └───┘ └───┘
└───┘ └───┘ └───┘
└───┘ └───┘ └───┘
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
60 package(s) needed for security, out of 77 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-9-130 ~]$ ls
last44.ipynb  last5.ipynb  mongo-spark  spark-3.4.0-bin-hadoop3.tgz  sparkk
[ec2-user@ip-172-31-9-130 ~]$ cd sparkk
[ec2-user@ip-172-31-9-130 sparkk]$ ls
bin      done44.py  kubernetes  licenses  python     RELEASE    Untitled.ipynb
conf     examples  last44.ipynb  logs      R          sbin       work
data     jars      LICENSE     NOTICE   README.md  untitled   yarn
[ec2-user@ip-172-31-9-130 sparkk]$ cd bin
[ec2-user@ip-172-31-9-130 bin]$ ls
beeline                pyspark                spark-class.cmd        spark-shell.cmd
beeline.cmd            pyspark2.cmd          spark-connect-shell     spark-sql
docker-image-tool.sh   run-example           sparkR                  spark-sql2.cmd
find-spark-home        run-example.cmd       sparkR2.cmd            spark-submit
find-spark-home.cmd   spark-class           sparkR.cmd              spark-submit2.cmd
load-spark-env.cmd     spark-class2.cmd      spark-shell             spark-submit.cmd
load-spark-env.sh      spark-shell2.cmd
[ec2-user@ip-172-31-9-130 bin]$ cd ..
[ec2-user@ip-172-31-9-130 sparkk]$ cd sbin
[ec2-user@ip-172-31-9-130 sbin]$ ls
decommission-slave.sh  start-worker.sh
decommission-worker.sh start-workers.sh
slaves.sh             stop-all.sh
spark-config.sh       stop-connect-server.sh
spark-daemon.sh       stop-history-server.sh
spark-daemons.sh     stop-master.sh
start-all.sh         stop-mesos-dispatcher.sh
start-connect-server.sh stop-mesos-shuffle-service.sh
start-history-server.sh stop-slave.sh

```

Figure 5.15 Utilize Spark in the Cloud

By utilizing distributed systems, the computational capacity expands in a horizontal manner, enabling the system to effortlessly handle extensive datasets. In order to achieve enhanced performance, it was necessary to use strategies and techniques that optimize productivity and effectiveness. Parallel processing was a fundamental attribute of distributed systems and, Spark playing a crucial role in the attainment of effective data processing. Instead of depending on a one machine to carry out all operations, parallelism entails breaking down jobs into smaller sub-tasks that can be simultaneously executed on nodes within the cluster.

The master node in this environment can be accessed by port 8080, while the worker nodes can be accessed through ports 8081, 8082, and 8083, respectively. The execution of the master occurs through: `./sbin/start-master`.

In this instance, the master has been executed, and access to the web interface was available through port 8080. Within this interface, the status of the master and the associated workers was displayed, as depicted in Figure 5.16.

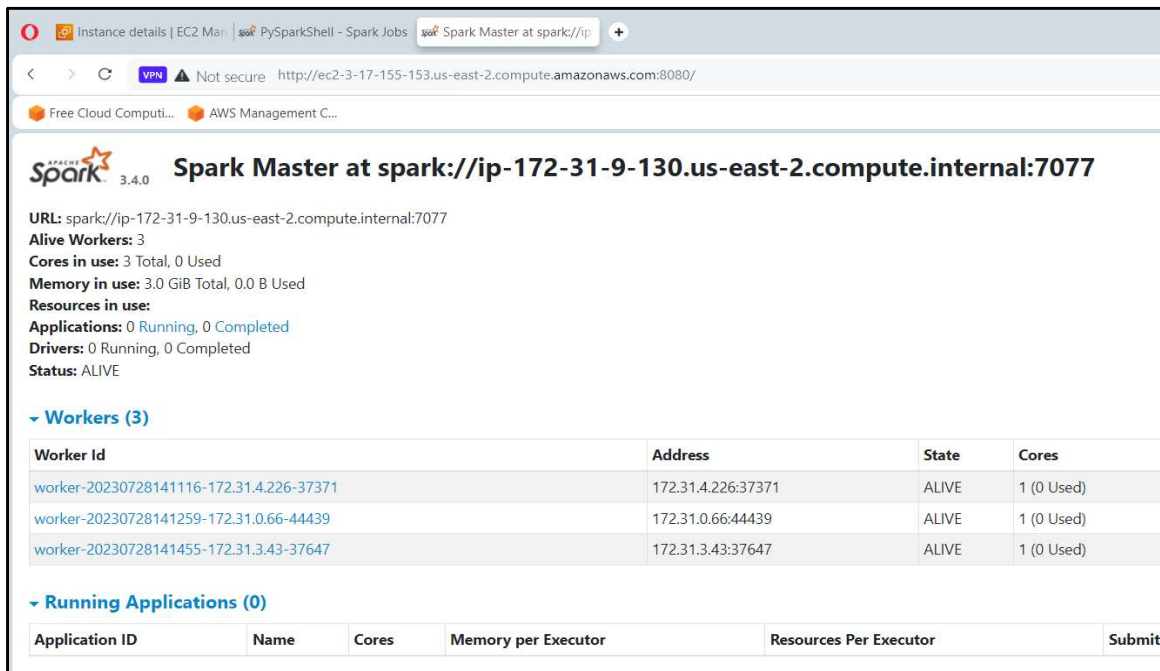


Figure 5.16 Launch the Workers and Master nodes

When running each of the worker nodes, the public DNS for each of them will be utilized, and the URL address of the master will be added, as follows:

`./sbin/start-worker.sh + URL in the master+7077.`

It concludes with port 7077, which is the default port for communication with the master. The worker is initiated and connected to the master operating within the same cluster, with a precaution to ensure that communication between devices was allowed by specifying inbound and outbound traffic. The required port was opened by configuring all machines to use the same security group. Similarly, Worker2 and Worker3 are activated, resulting in a page containing all the workers linked to the master via Port 8080 as show in figure 5.16.

In this scenario, reporting was sent from these workers to the master. When *Spark submit* is executed, the request is processed as a class file inside a JAR file, as follows: *`./bin/spark-submit`*. Also, the operational status of Jupyter Lab was verified by navigating to the following URL: *`http://ip4(master):8888`*

Therefore, parallel processing a technique was used substantially to decrease the amount of time needed to handle large datasets. The distributed nature of the cluster allows for parallel processing of data by individual worker nodes,

resulting in a significant decrease in overall processing time and enabling scalability as the dataset expands. This guarantees the maintenance of optimal processing efficiency, even in the face of escalating data quantities. Also, Parallel processing was used to enhance the efficiency of resource allocation. Every node plays a role in the processing work, effectively utilizing the available CPU, memory, and storage resources.

The implementation of the proposed approach in a cloud environment presents several notable benefits. These advantages encompass the ability to dynamically scale resources to accommodate the growth of datasets, the utilization of cost-efficient pay-as-you-go models, the effective utilization of parallelism to enhance processing speed, the availability of simplified management tools, global accessibility, data redundancy, and seamless integration with diverse services.

5.3.4 Procedure for Data Processing Output

In the data processing workflow with Apache Spark, the first step involves data retrieval from MongoDB Atlas, where the researchers' information is stored. Once the data is brought into a Spark DataFrame, various operations can be performed on it. This is initiated by selecting relevant columns, such as names, affiliations, and email addresses. Then filtering the data to focus on specific subsets, for instance, researchers with a certain number of citations. Summarizing and analyzing data through grouping, aggregation operations allow statistics like the total number of citations or the average h-index to be calculated. In cases where there are multiple DataFrames or related data sources, they can be joined to create a unified dataset. Data sorting aids in the identification of top researchers or the arrangement of data. The essential data processing steps in Apache Spark include the handling of duplicates and empty values. The removal of duplicates in the dataset was crucial as it can skew analysis results and lead to incorrect conclusions. The `'dropDuplicates()'` method provided by Spark enables the identification and elimination of duplicate records based on one or more columns. In the context of researchers' data, this ensures that each researcher was represented only once, thereby enhancing data accuracy and consistency.

When dealing with empty or missing values, often denoted as null or NaN, in real-world datasets, Spark offers various mechanisms for addressing them. The `'dropna()'` method is used to remove rows containing empty values, which is particularly useful when missing data is irrelevant to the analysis. Additionally, the `'fillna()'` method is employed to replace missing values with a default value, preserving data integrity.

By these data processing techniques being incorporated into the Apache Spark workflow, it is ensured that the data is clean, made complete, and prepared for accurate analysis. Finally, the processed data was saved back to MongoDB Atlas, making it ready for further analysis. This comprehensive data processing approach in Apache Spark facilitates the efficient and effective transformation and analysis of researchers' data. Figure 5.17a, 5.17b represent the sample of data after processing.

	A	B	C	D	E	F	G	H
	indx	name	real affiliation	email	citations	h-index	i10 index	interest
1	800003	Fatima S. Hussein	University of babylon	student.uobabylon.edu.iq	65094	121	334	Fluids
5	800004	Faris Al Lami	University of baghdad	comed.uobaghdad.edu.iq	54889	44	73	Public Health
6	800005	Alaa BA Baban	Bayan University	bnru.edu.iq	54166	110	266	كشور ورجعة
7	800006	عادل كافي الزبيدي	University of baghdad	coadec.uobaghdad.edu.iq	51550	82	156	تكنولوجيا المعلومات على المستوى الإداري الإدارة المحلية الإدارة
8	800007	mahmood	University of Mosul	uomosul.edu.iq	36047	83	275	mathematics
9	800008	Shuab jassim	University of Anbar	uonbar.edu.iq	33856	80	356	المطالعة
10	800009	S.Mohammad Sajadi	Cihan University	chihanniversity.edu.iq	32991	71	177	Phytochemistry
11	800010	Sarab Dalaf	University of Tikrit	tu.edu.iq	25091	24	26	genotoxicology
12	800011	Richard D. Handy	Cihan University	chihanniversity.edu.iq	22267	68	153	Environmental Toxicology and Human Health
13	800012	Bayan Alhanea	Al-Manara College for Medical Science	uomanara.edu.iq	22076	59	103	Removable Prosthodontics
14	800013	Tariq Albshtir	University of Tikrit	tu.edu.iq	20452	73	272	الطب الترميمية
15	800014	مروية عبد الصالح	University of Tikrit	tu.edu.iq	19361	50	128	الرياضيات
16	800015	Fatima Shbas	University of Tikrit	st.tu.edu.iq	17132	64	266	clinical biochemistry
17	800016	Huda M. Ahmed	University of Diyala	ecomang.uodiyala.edu.iq	14945	2	1	علوم رياضيات
18	800017	jaber Qasim Jaber	Al-Zahravi University College	g.alzahr.edu.iq	14814	62	326	pediatric nursing
19	800018	jawad Rustum	University of kirkuk	uokirkuk.edu.iq	14025	56	125	Reservoir Engineering
20	800019	Abdul Amir H. Kadhum	University of Al-Ameed	alameed.edu.iq	13260	62	227	Chemistry and Chemical Engineering
21	800020	Ali Hussain Rastlak	University of Basrah	uobasrah.edu.iq	12935	52	319	Laser design
22	800021	Andrea Peroni	Tisik International University	tiu.edu.iq	12608	63	159	ethnobiology
23	800022	Ali Alhanzawi	University of Al-Qadisiyah	qu.edu.iq	12171	41	65	Psychiatry
24	800023	Ali Esmail Al-Snafi	Tiqar University	utq.edu.iq	12130	63	246	Pharmacology
25	800024	abbas kadhim	University of kirkuk	uokirkuk.edu.iq	11784	60	209	
26	800025	Majid Sakhi Jabir	University of technology	uctechnology.edu.iq	11667	33	52	Immunology
27	800026	Ahmed	Hilla University College	hillaunc.edu.iq	10927	62	217	Medical Microbiology
28	800027	Mahmood Ameen Abdulla	Cihan University	chihanniversity.edu.iq	10763	61	214	Histology
29	800028	Ahmed Idan jasm	University of Anbar	uonbar.edu.iq	10050	60	145	فسيولوجيا الجهاز الادمي
30	800029	Prof. Magdam Tariq Charchar	University of technology	uotechnology.edu.iq	9434	57	175	Renewable and sustainable energy
31	800030	biha meshthen	Al-Hajai University	alhajiaia.edu.iq	8628	27	365	لغة عربية
32	800031	Aso Darvesh	University of Human Development	uhd.edu.iq	8489	26	40	Serious Games
33	800032	Nora. Rashid	Northern Technical University	ntu.edu.iq	8248	35	109	computer engineering
34	800033	Mustafa Amwer Hussain	Alfarabi University college	alfarabic.edu.iq	8041	43	81	Biology

Figure 5.17a, Sample of Authors DataSet After Processing

A	B	C	D	E	F	G	H
index	name	paper	article_title	year	co-authors	article_publication	article_cited_by_value
800003	Fatima S. Hussein	0	Personalizing the treatment of women with early breast cancer: high	2013	A Goldhirsch, EP Winer, AS Coates, RD Gelber, M Piccart	Annals of oncology 24 (9), 2206-2223, 2013	3574
800003	Fatima S. Hussein	1	Gene expression profiling in breast cancer: understanding the molecu	2006	C Sotirou, P Waparat, S Loi, A Harris, J Smeets, J Journal of the National Cancer Institute 98 (4),		2136
800003	Fatima S. Hussein	2	Taioring therapies—improving the management of early breast cancer	2015	AS Coates, EP Winer, A Goldhirsch, RD Gelber, M Grant	Annals of oncology 26 (8), 1533-1546, 2015	1895
800003	Fatima S. Hussein	3	Primary breast cancer: ESMO Clinical Practice Guidelines for diagnc	2015	E Sertkus, S Kyriakides, S Ohno, F Penault-Llorca, P Po	Annals of oncology 26 (suppl. 5), v6-x30, 2015	1701
800003	Fatima S. Hussein	4	70-gene signature as an aid to treatment decisions in early-stage bre	2016	F Cardoso, L Vant Veer, J Bogerits, L Slaets, G Viale, S	New England Journal of Medicine 375 (8), 717	1485
800003	Fatima S. Hussein	5	Validation and clinical utility of a 70-gene prognostic signature for woi	2006	M Bysse, S Loi, L Vant Veer, G Viale, M Delorenzi, AM	Journal of the National Cancer Institute 98 (17)	1436
800003	Fatima S. Hussein	6	Early breast cancer: ESMO Clinical Practice Guidelines for diagnosi	2019	F Cardoso, S Kyriakides, S Ohno, F Penault-Llorca, P	Annals of Oncology 30 (8), 1194-1220, 2019	1271
800003	Fatima S. Hussein	7	Endocrine therapy plus zoledronic acid in premenopausal breast can	2009	M Grant, B Milneritsch, W Schipinger, G Luschin-Ebent	New England Journal of Medicine 360 (7), 679	1208
800003	Fatima S. Hussein	8	Endocrine therapy plus zoledronic acid in premenopausal breast can	2009	M Grant, B Milneritsch, W Schipinger, G Luschin-Ebent	New England Journal of Medicine 360 (7), 679	1208
800003	Fatima S. Hussein	9	Ki-67 as prognostic marker in early breast cancer: a meta-analysis o	2007	E De Azavedo, F Cardoso, G de Castro, M Colozza, M	British journal of cancer 96 (10), 1504-1513, 2	1123
800003	Fatima S. Hussein	10	Eribulin monotherapy versus treatment of physician's choice in patier	2007	J Cortes, J O'Shaughnessy, D Loesch, J Blum, LT Vaid	The Lancet 377 (9769), 914-923, 2011	1113
800003	Fatima S. Hussein	11	Strong time dependence of the 76-gene prognostic signature for node	2007	C Desmedt, F Pielte, S Loi, Y Wang, F Lalleu, B Halber	Clinical cancer research 13 (11), 3207-3214, 2	1061
800003	Fatima S. Hussein	12	Pembrolizumab for early triple-negative breast cancer	2020	P Schmid, J Cortes, L Pusztai, H McArthur, S Kimmell, J	New England Journal of Medicine 382 (9), 810	966
800003	Fatima S. Hussein	13	Correction to: De-escalating and escalating treatments for early-stage	2018	G Curigliano, HJ Burstein, EP Winer, M Grant, P Dubsy	Annals of Oncology 29 (10), 2153, 2018	969
800003	Fatima S. Hussein	14	4th ESO-ESMO International Consensus Guidelines for Advanced Bl	2018	F Cardoso, E Sertkus, A Costa, E Papadopoulos, M Aap	Annals of Oncology 29 (8), 1634-1651, 2018	965
800003	Fatima S. Hussein	15	De-escalating and escalating treatments for early-stage breast cancer	2017	G Curigliano, HJ Burstein, E P Winer, M Grant, P Dubsy	Annals of Oncology 28 (8), 1700-1712, 2017	963
800003	Fatima S. Hussein	16	American society of clinical oncology clinical practice guideline upda	2010	HJ Burstein, JJ Gignis, AA Prestud, S Terin,	Journal of oncology practice 6 (5), 243-246, 20	933
800003	Fatima S. Hussein	17	Cardiovascular toxicity induced by chemotherapy, targeted agents ar	2012	G Curigliano, D Cardinale, T Suter, G Platanotis, E De A	Annals of Oncology 23 (suppl. 7), vi155-vi166	907
800003	Fatima S. Hussein	18	Adjuvant exemestane with ovarian suppression in premenopausal bre	2014	O Pagan, MM Regan, BA Walely, GF Fleming, M Collea	New England Journal of Medicine 371 (2), 107	790
800003	Fatima S. Hussein	19	European School of Oncology	2010	F Amant, S Deckers, K Van Calsteren, S Lohd, M Halasz	European journal of cancer 46 (18), 3150-3166	765
800003	Fatima S. Hussein	20	3rd ESO-ESMO international consensus guidelines for advanced bre	2017	F Cardoso, A Costa, E Sertkus, M Aapro, F ie, CH Barni	Annals of Oncology 28 (1), 16-33, 2017	714
800003	Fatima S. Hussein	21	ESO-ESMO 2nd international consensus guidelines for advanced bre	2014	F Cardoso, A Costa, L Norton, E Sertkus, M Aapro, F ie	Annals of Oncology 25 (10), 1871-1888, 2014	769
800003	Fatima S. Hussein	22	Adjuvant ovarian suppression in premenopausal breast cancer	2015	PA Francis, MM Regan, GF Fleming, L Liang, E Crieuols,	New England Journal of Medicine 372 (5), 436	738
800003	Fatima S. Hussein	23	A new molecular predictor of distant recurrence in ER-positive, HER2	2011	M Fillois, M Rudes, R Jakasz, P Dubsy, F Fitza, CF S	Clinical Cancer Research 17 (18), 5012-5020,	666
800003	Fatima S. Hussein	24	Primary breast cancer: ESMO Clinical Practice Guidelines for diagnc	2013	E Sertkus, S Kyriakides, F Penault-Llorca, P Poortmans,	Annals of Oncology 24 (suppl. 6), vi7-423, 201	672
800003	Fatima S. Hussein	25	Long-term outcomes for neoadjuvant versus adjuvant chemotherapy in	2018	B Asselain, W Barlow, J Bartlett, J Beigh, E Bergsten	The Lancet Oncology 19 (1), 27-39, 2018	627
800003	Fatima S. Hussein	26	Adjuvant chemotherapy in older women with early-stage breast cance	2009	HB Muss, DA Berry, CT Cirincione, M Theodorou, AM	New England Journal of Medicine 360 (20), 20	587
800003	Fatima S. Hussein	27	Locally recurrent or metastatic breast cancer: ESMO Clinical Practic	2012	F Cardoso, N Harbeck, L Falloisfeld, S Kyriakides, E S	Annals of oncology 23 (suppl. 7), vi11-vi19, 21	576
800003	Fatima S. Hussein	28	Overexpression of hypoxia-inducible factor 1 α is associated with an l	2002	M Schindl, SF Schoppmann, H Samonigg, H Hausmanig	Clinical Cancer Research 8 (6), 1831-1837, 20	568
800003	Fatima S. Hussein	29	Phase II study of temsirolimus (CCI-779), a novel inhibitor of mTOR, i	2005	S Chan, ME Schenulen, S Johnston, K Moss, F Cardos	Journal of Clinical Oncology 23 (23), 5314-532	555
800003	Fatima S. Hussein	30	Clinical application of the 70-gene profile: the MINDACT trial	2008	F Cardoso, L Vant Veer, E Rutgers, S Loi, S Mook, M	J Journal of clinical oncology 26 (5), 729-735, 20	549
800003	Fatima S. Hussein	31	Adjuvant endocrine therapy plus zoledronic acid in premenopausal w	2011	M Grant, B Milneritsch, H Stoeger, G Luschin-Ebengru	The lancet oncology 12 (7), 631-641, 2011	532
800003	Fatima S. Hussein	32	Neratinib after trastuzumab-based adjuvant therapy in HER2-positive	2017	M Martin, FA Holmes, B Eljertsen, S Dalaloe, B Moy, H	The Lancet Oncology 18 (12), 1688-1700, 201	513
800003	Fatima S. Hussein	33	Neratinib after trastuzumab-based adjuvant therapy in patients with H	2016	A Chan, S Dalaloe, FA Holmes, B Moy, H Wata, VJ Har	The Lancet Oncology 17 (3), 367-377, 2016	509
800003	Fatima S. Hussein	34	High risk of recurrence for patients with breast cancer who have hum	2009	AM Gonzalez-Angulo, JK Litton, KR Broglio, F Merc-B	Journal of clinical oncology 27 (34), 5700, 200	497
800003	Fatima S. Hussein	35	Multidisciplinary meeting on male breast cancer: summary and rese	2010	LA Korde, JA Zujewski, L Kamin, S Giordano, S Dornic	Journal of Clinical Oncology 28 (12), 2114, 20	489
800003	Fatima S. Hussein	36	Adjuvant denosumab in breast cancer (ABCOSG-18): a multicentre, ra	2015	M Grant, G Pielier, PC Dubsy, M Hubalek, R Grell, R	Le Lancet 386 (9992), 433-443, 2015	479
800003	Fatima S. Hussein	37	Clinical use of biomarkers in breast cancer: Updated guidelines from	2017	MJ Duffy, N Harbeck, M Nap, R Molina, A Nicolini, E	Sen European journal of cancer 75, 284-298, 2017	475
800003	Fatima S. Hussein	38	Assessing tumor infiltrating lymphocytes in solid tumors: a practical	2017	S Hendry, R Sagaido, T Geraert, PA Russell, T John, B	T Advances in anatomic pathology 24 (6), 311, 2	453
800003	Fatima S. Hussein	39	International guidelines for management of metastatic breast cancer:	2010	O Pagan, E Sertkus, W Wood, M Collea, T Cifer, S Ky	Journal of the National Cancer Institute 102 (7)	433
800003	Fatima S. Hussein	40	Taioring adjuvant endocrine therapy for premenopausal breast cancer	2018	PA Francis, O Pagan, GF Fleming, BA Walely, M Collea	New England Journal of Medicine 379 (2), 122	431

Figure 5.17b, Sample of Articles DataSet After Processing

5.3.5 Performance Evaluation

Performance evaluation in this context encompasses two distinct types, each serving a specific purpose. The first is the *Iraqi Scholar Network Performance Evaluation* (ISN), which delves into assessing the performance and effectiveness of the Iraqi Scholar Network. The second, known as *Iraqi University Network Performance Evaluation* (IUN), focuses on evaluating the performance and functionality of the Iraqi University Network. These two evaluation methodologies play crucial roles in Measuring and improving the efficiency of their respective networks, contributing to their continuous development and enhancement.

5.3.5.1 ISN Performance Evaluation

In this section, the evaluation of the ISN network was performed based on the authors' level and as following:

A. Number of Published Papers

According to the collected data of the Iraqi authors from Google Scholar, descriptive statistics were extracted from the data. Figure 5.18 shows the frequency of published papers by Iraqi authors. The Figure also shows that the number of published papers is grouped by universities.

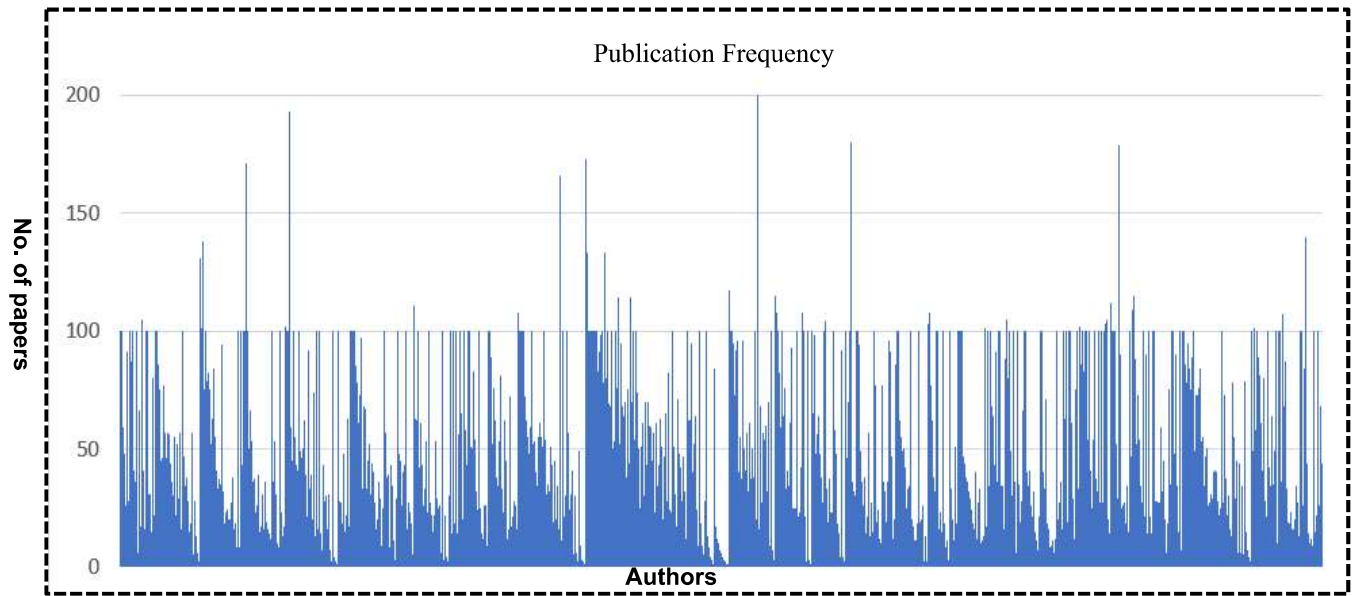


Figure 5.18 The frequency of papers published by Iraqi authors (group by universities)

The main observation shown is that the pattern of all Iraqi universities follows the same power-law distribution. In other words, in each university, there are few authors with a high number of publications, while the majority of authors have few publications. This pattern reflects the fact that the frequency of publications depends on a few authors in a university. The origin of this interpretation is based on the Pareto rule (see Figure 5.19), which states that for any phenomenon that follows a power-law distribution, there exists 80% of a population driven by the other 20% of activities.

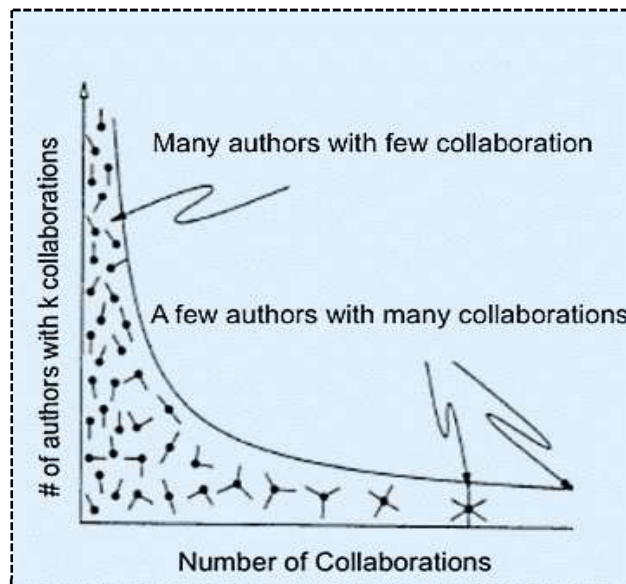


Figure 5.19 Pareto Rule Interpretation Regarding the Iraqi Authors.

B. Citation

The frequency of citations is also extracted from the collected dataset of the Iraqi authors. Figure 5.20 demonstrates the performance of the Iraqi authors in term of citations. It can be observed that the pattern of citations was similar to the pattern observed in the number of publications of authors.

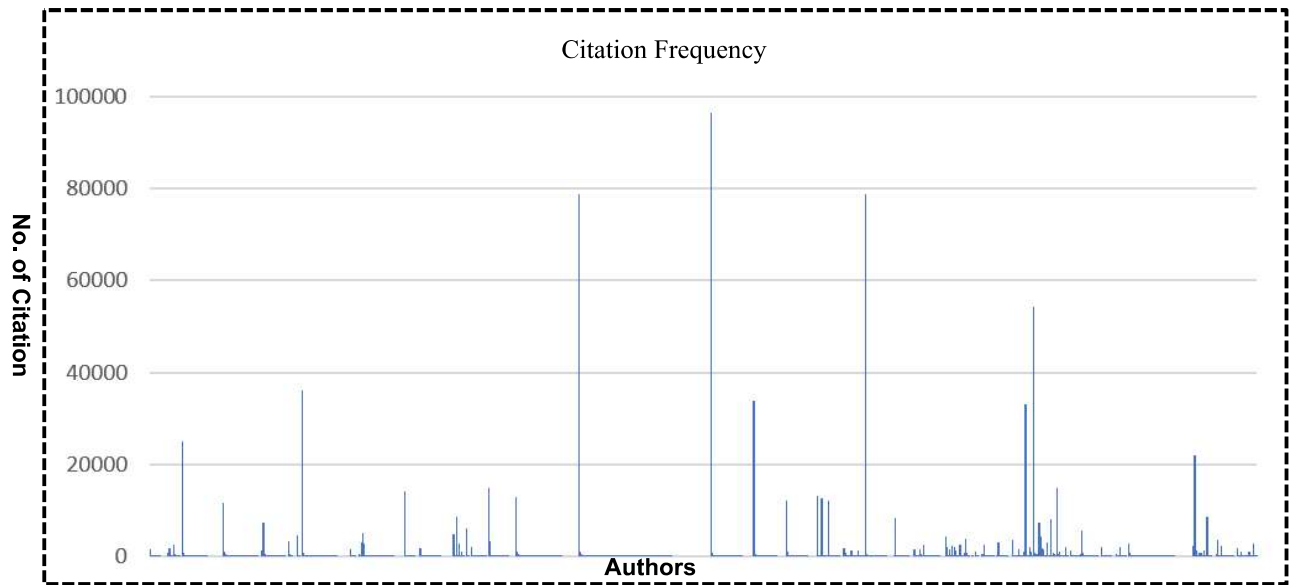


Figure 5.20 The frequency of citations of the Iraqi authors (group by universities)

C. H-Index and I10-Index

The same pattern shown in the previous figures (5.18 and 5.20) is observed when extracting the performance of the h-index and i10 index of the Iraqi authors (see Figures 5.21 and 5.22). This result confirms the fact stated in Figure 5.19, that a few Iraqi authors have the most significant impact on the academic performance of the whole performance of the Iraqi scholars.

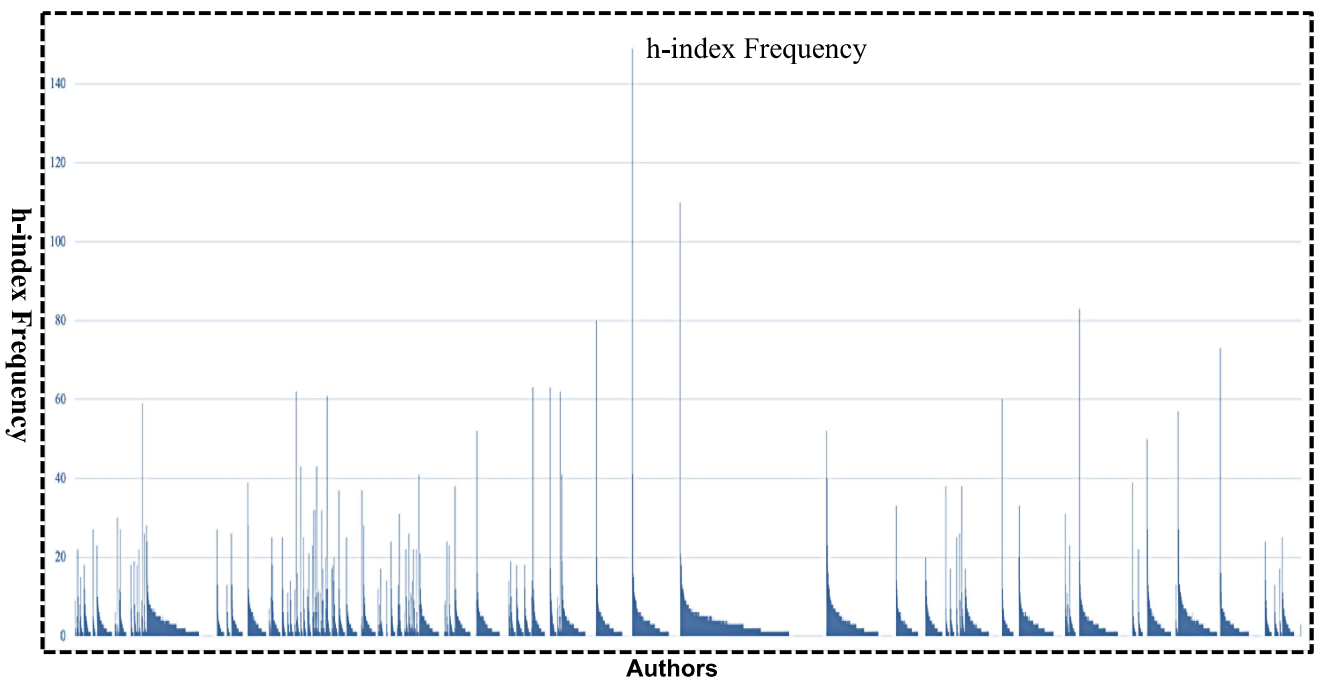


Figure 5.21 The Frequency of the h-index of the Iraqi Authors (Grouped by Universities)

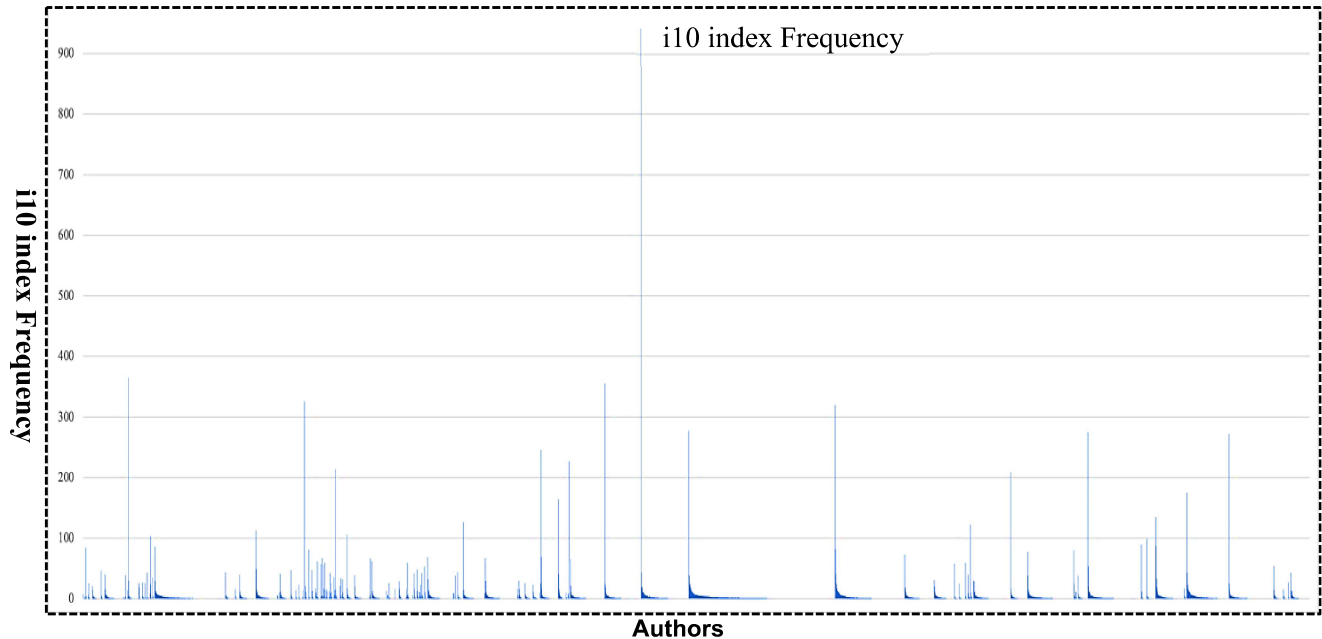


Figure 5.22 The Frequency of the I10-index of the Iraqi Authors (Grouped by Universities)

D. ISN Network Visualization

It is required, furthermore to use the collected dataset and the concepts of complex networks to visualize Iraqi Scholars Network (ISN) and show the main facts and features. Figure 5.23 shows the visualization of the network, each node in the network represents an author and an edge created between two authors when coauthoring a paper. Different colors of cluster represent different universities. The network shows many clusters, each of which represents a university. The biggest cluster represents the international authors while the other small clusters represent the local Iraqi scholars. It is clear that there is a high level of collaboration between local and international authors. It is believed that this collaboration is mainly a side effect of the Iraqi scholarships and the students who study abroad. In fact, more investigations are required to analyze the network, therefore, network measurements were used.

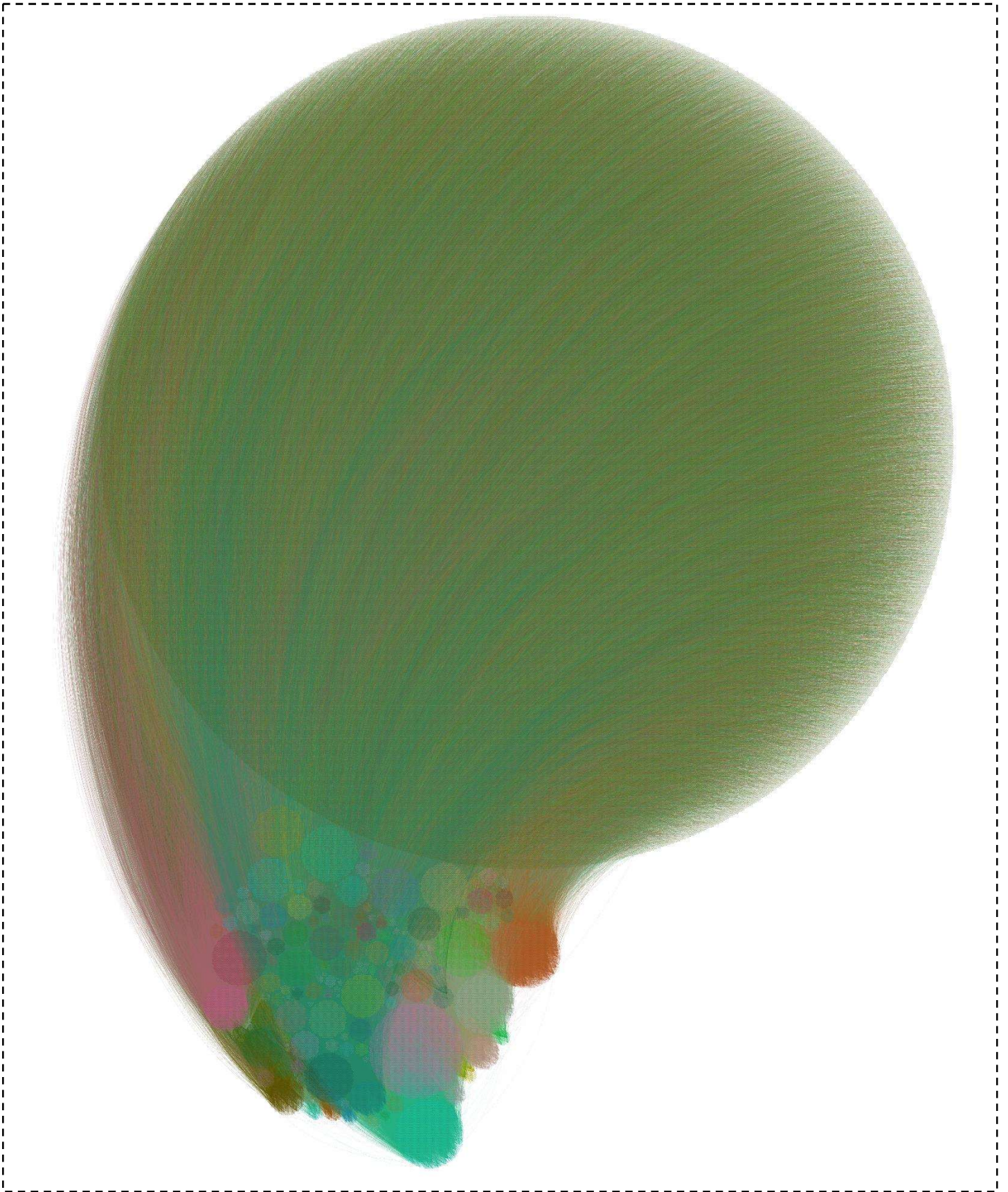


Figure 5.23 Visualization of ISN Network of Local and International Scholars

E. Main Characteristics

As mentioned in the previous chapter, two networks were created in this thesis, the first Network is for authors while the second is for the universities. The first network represents the entire network of papers coauthored with Iraqi scholars by local and international authors. The characteristics of the authors' network are presented in Table 5.2, which shows that the average degree of coauthoring papers is 1.805. This means each author in the Iraqi universities has collaborated on approximately two papers (local, international, or both). Diameter of the network reflects 6 steps of distance to get between the farthest two authors in the network. An average shortest path was shown to be 2.111, which suggests that, on average authors are reasonably well-connected, with a relatively short path between them. In summary, the network appears to have a moderate level of collaboration, reasonable connectivity, and efficient paths for collaboration between authors.

Table 5.2: Iraqi Scholars Network

Number of Nodes (Authors)	Number of Edges (Coauthorship)	Average Degree	Network Diameter	Average Shortest Path Length
423410	382142	1.805	6	2.111

F. Benchmarking

Claiming acceptability of results means that the results obtained were compared to other similar networks in the literature as shown in Table 5.3. The generated network for Iraqi authors was benchmarked with two similar networks in the literature. Table 5.3 shows that the performance of the Iraqi network of authors has close values in terms of the average clustering coefficient and the average shortest path length. In fact, the reason behind using these metrics is that they can be used to compare two networks as a whole. The average clustering coefficient is used to measure the tendency of authors to collaborate with each other which shows a low tendency of the authors to collaborate with each other. The other metric is Average Shortest Path Length, which reflects the shortest path between authors to collaborate within the network. The value 2.111 is considered acceptable compared to the ACM authors network, which means there exist authors who play the role of a bridge between the communities of authors, and this is clear when observing the

Average Clustering Coefficient. This metric is proven efficient in measuring the performance of a network.

Table 5.3: Comparison Between the ISN and Other Similar Networks in the Literature

Network	Average Degree	Average Clustering Coefficient	Average Shortest Path Length
ISN (Iraqi Scholars Network)	1.805	0.249	2.111
ACM Authors Network	2.9	0.6	4.99
ACM Physics Network	2.7	0.362	3.38

G. ISN Main Centrality Metrics

The network centrality measurements used can be summarized as follows:

- **Betweenness Centrality:** Figure 5.24 depicts the levels of betweenness centrality of the authors of different universities. The figure also shows that the majority of authors have low levels of betweenness, which means the ISN has a few numbers of influential scholars.

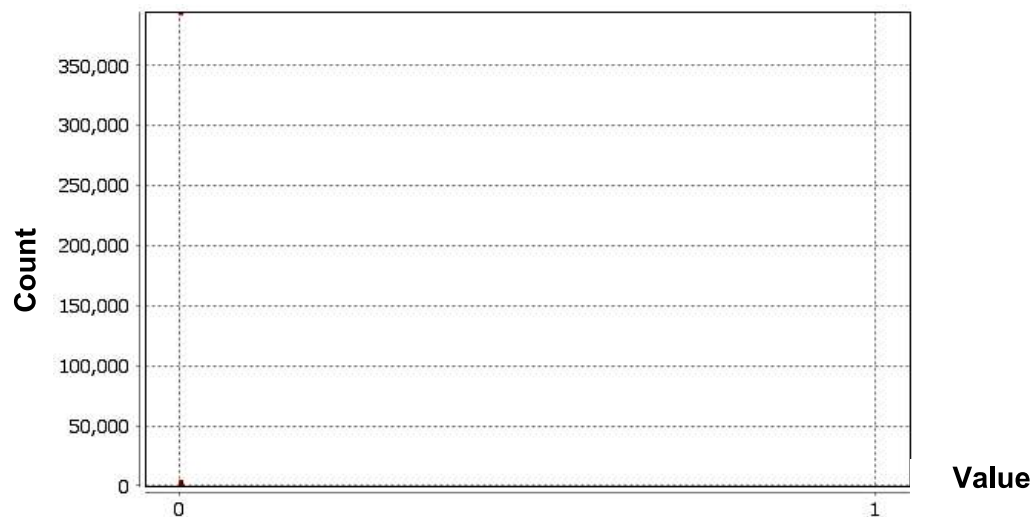


Figure 5.24 Distribution of Betweenness Centrality

- **Degree Centrality:** Figure 5.25 depicts the frequency of coauthoring papers of the authors of different universities. As observed from the figure, the distribution of degrees follows a power law. This fact proves that the majority of authors have few collaborations in terms of coauthoring

papers, while few authors have a high frequency of collaboration. This fact follows what has been mentioned about Pareto rule.

- **Closeness Centrality:** Figure 5.26 depicts the levels of closeness centrality of authors of different universities. It can be observed that most of the levels are between 0.5 and 0.7, which reflects the fact that authors in ISN are relatively close to each other. However, according to the dataset, it is observed that the closeness level is a side effect of the collaboration of authors of the same university with the existence of a few authors who collaborated with authors from other universities.

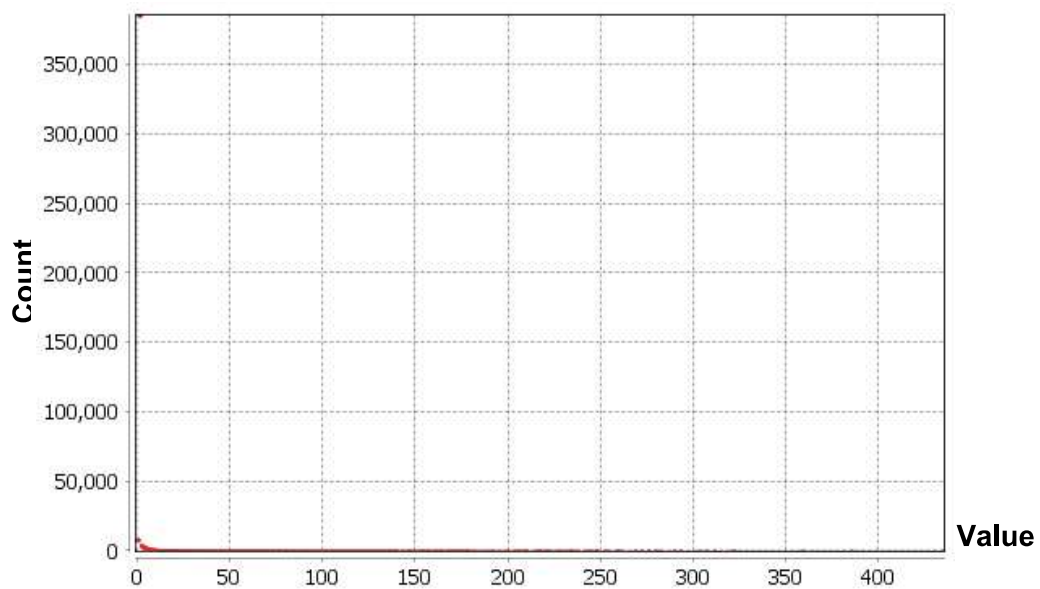


Figure 5.25 Distribution of Degree Centrality

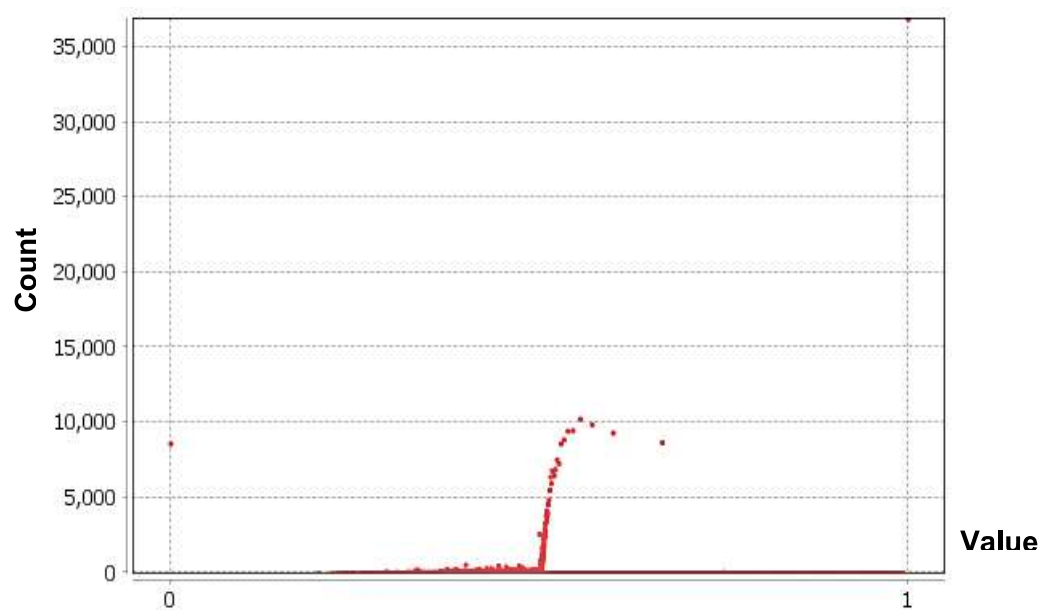


Figure 2.26 Distribution of Closeness Centrality

H. ISN with No International Authors

By removing the international scholars from ISN, the visualization and the characteristics of the Iraqi Scholars Network (ISN) are reperformed. Table 5.4 presents the characteristics of the ISN after removing the international authors.

Table 5.4: ISN Network Characteristics After Removing the International Scholars

Number of Nodes	Number of Edges	Average Degree	Network Diameter	Average Path Length
43016	1863	0.087	4	1.245

Table 5.4 shows that the characteristics of the ISN have been changed compared to Table 5.2 which includes the international authors. The ISN with no international authors reflects the pure relations among the Iraqi authors. However, the performance is decreased when observing the values of average degree, diameter, and average path length. This is because some international authors play the role of a bridge. For instance, two Ph.D. students from two different Iraqi universities are supervised by a professor from a university that both students study in. The visualization of ISN that includes only Iraqi authors is shown in Figure 5.27.

The visualization of the network shows many clusters, each of which represents a university. Different colors show different universities, and the size of clusters reflects the size of universities in terms of the number of authors. The weight of the edges reflects the strength of the collaboration in terms of co-authoring papers. It can be observed that there are Iraqi universities which have strong collaboration in terms of co-authoring papers.

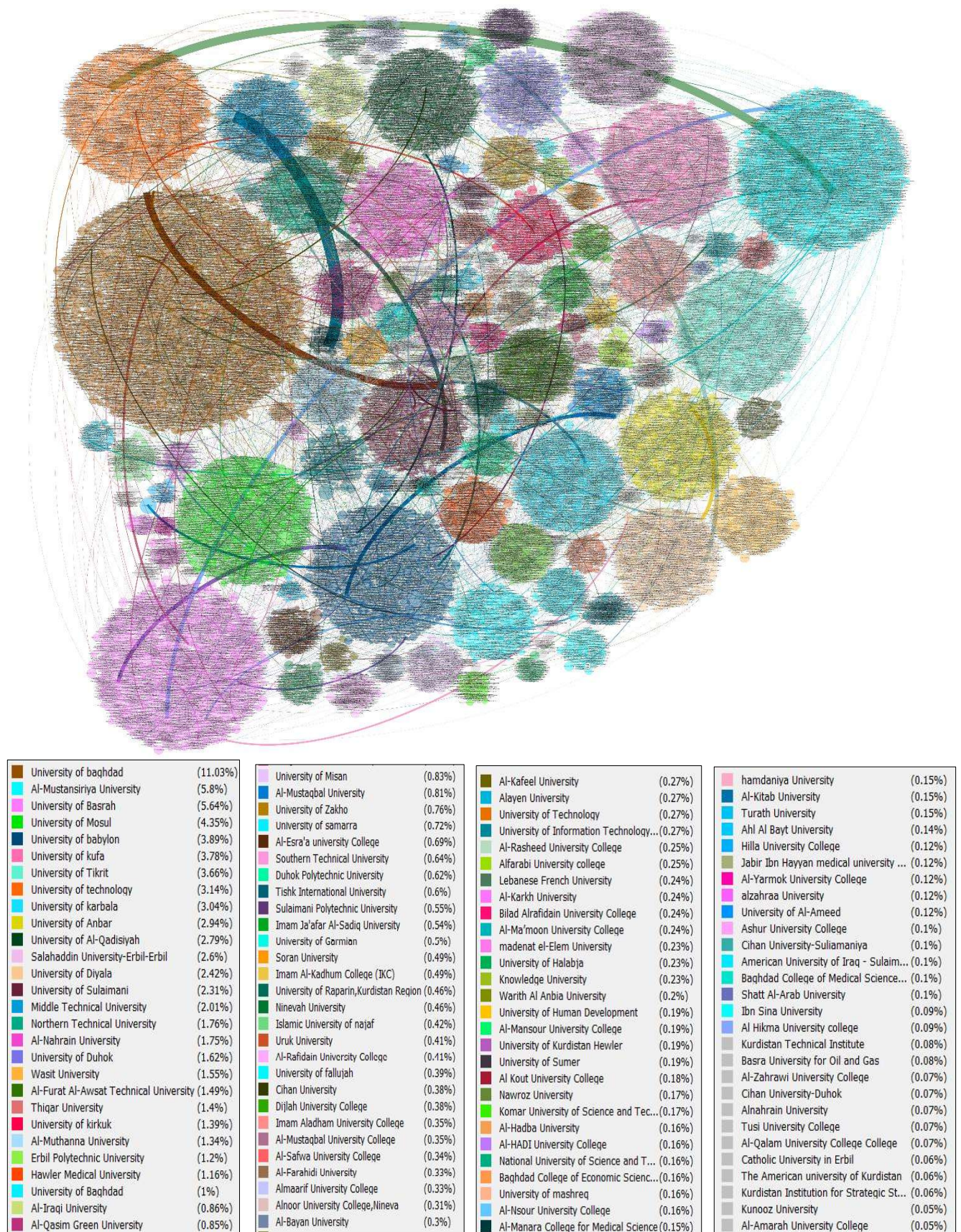


Figure 5.27 Visualization of ISN of Iraqi Authors Only

The centrality measurements of updated ISN are calculated such that the distribution of betweenness centrality is shown in Figure 5.28, which has the same behavior of the ISN with international authors.

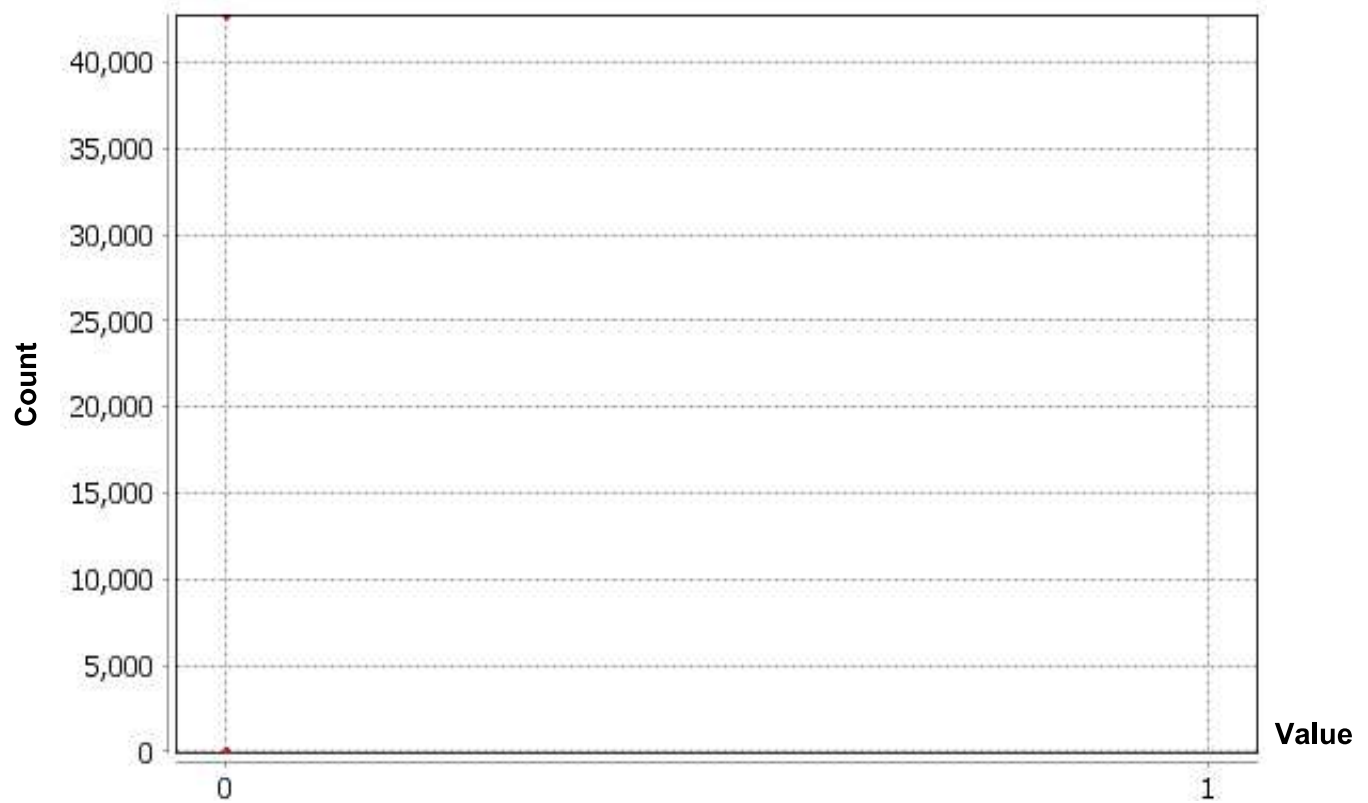


Figure 5.28 Betweenness Centrality Distribution

The degree distribution and the eigen centrality show a power law behavior, which is expected (see Figures 5.29, 5.30 and 5.31 respectively). The closeness centrality, however, has values that are mostly above 0.4, which means that the Iraqi authors are close to each other and this closeness is due to the collaboration of authors of the same university.

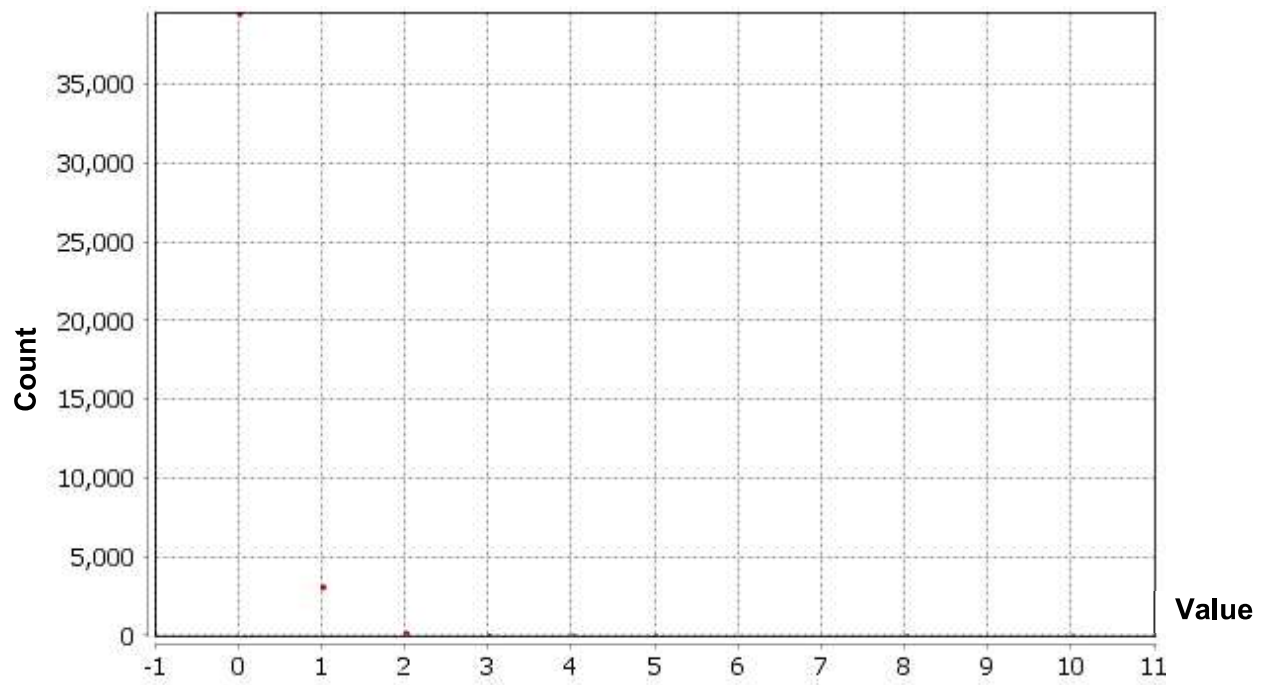


Figure 5.29 Degree Distribution Centrality

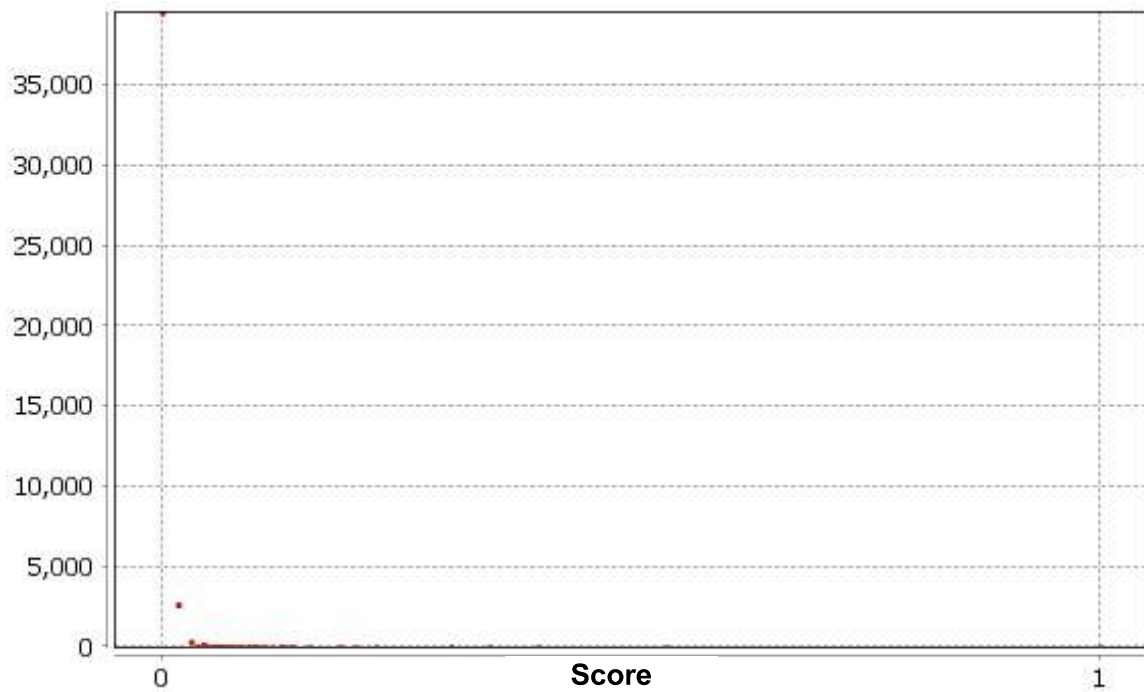


Figure 5.30 Eigen Centrality Distribution

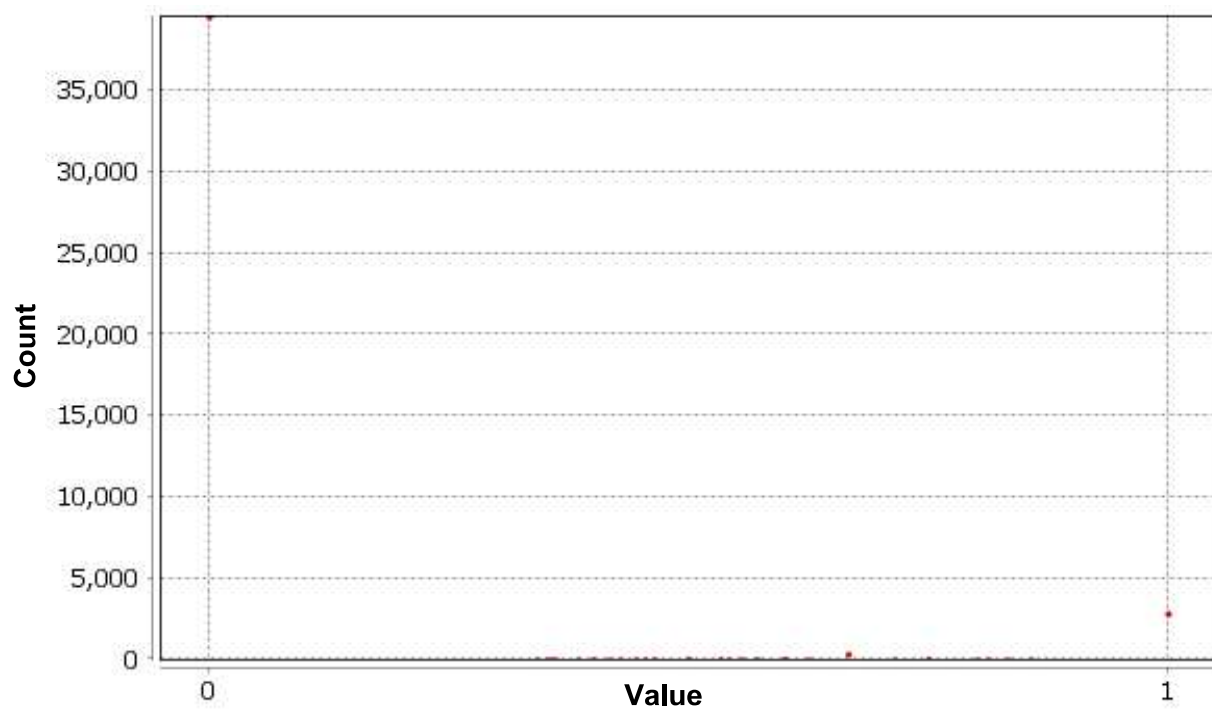
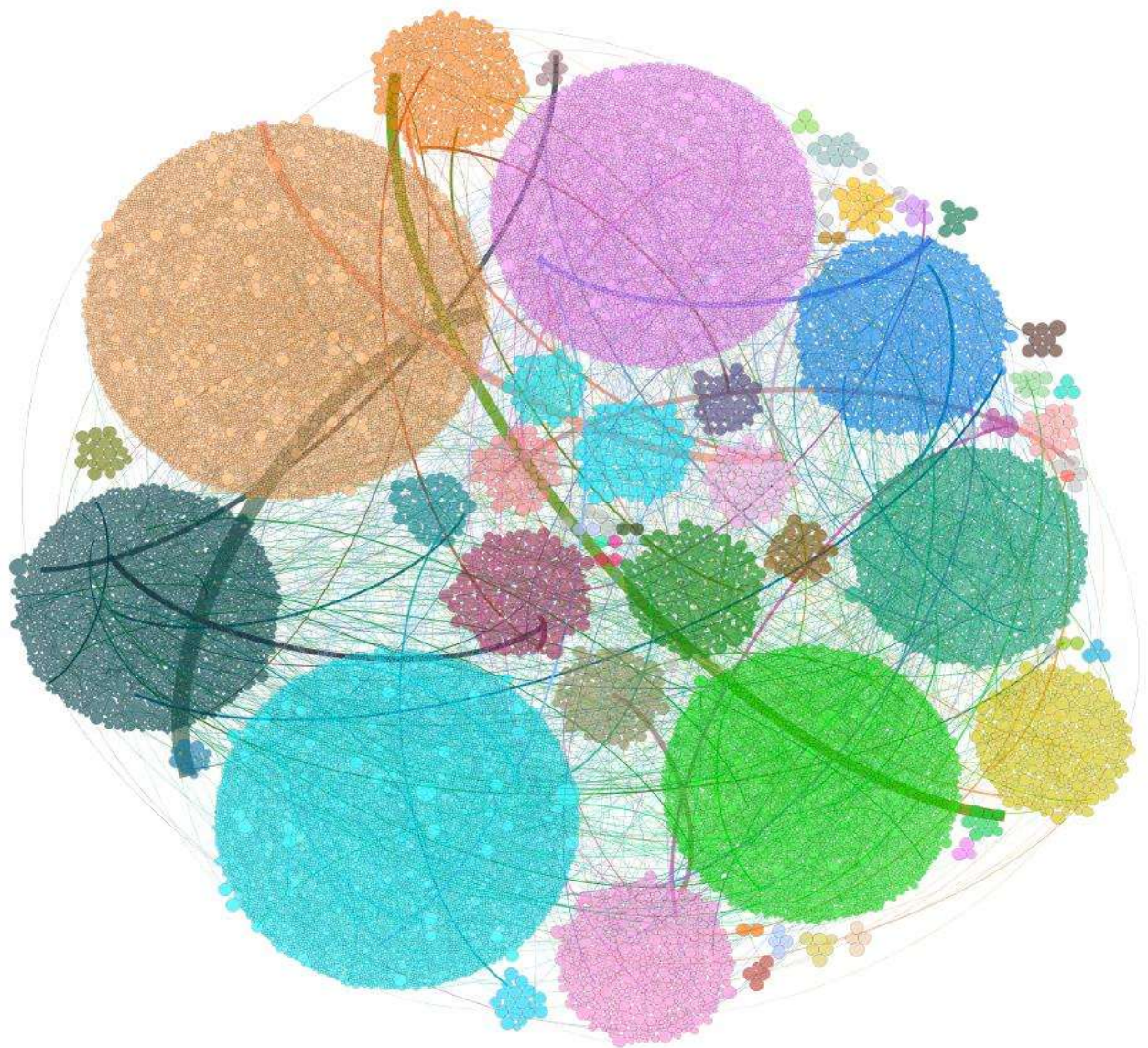


Figure 5.31 Closeness Centrality Distribution

I. ISN H-Index and I10-Index Evaluation

The performance of the h-index of the ISN is visualized in Figure 5.32. The interesting fact about this visualization is that the majority of the Iraqi authors have no h-index, which means that they have either no citations or they have few papers, which is a negative indicator. Also, the visualization shows that the authors with an h-index of 1 is significantly high followed by an h-index of 2, and so on. The authors with a high h-index are too low compared to the number of authors in the universities.

These results can be of interest to the Iraqi officials in the Ministry of higher education and scientific research, as these values can be further used as indicators or thresholds in the promotion procedures in Iraqi universities. Figure 5.33 depicts the performance of the ISN network authors in terms of i10- index. The same of the previous observations can be obtained when evaluating i10-index.



h-index	
orange	null (30.22%)
cyan	1 (25.88%)
purple	2 (15.78%)
green	3 (9.84%)
dark blue	4 (5.94%)
teal	5 (3.73%)
blue	6 (2.38%)
pink	7 (1.52%)
yellow	8 (1.01%)
orange	9 (0.81%)
dark red	10 (0.59%)
dark green	11 (0.5%)
light blue	12 (0.32%)
olive	13 (0.26%)
light pink	14 (0.18%)
red	15 (0.14%)
light cyan	16 (0.13%)
dark teal	17 (0.12%)

Figure 5.32 ISN Performance in term of h-index.

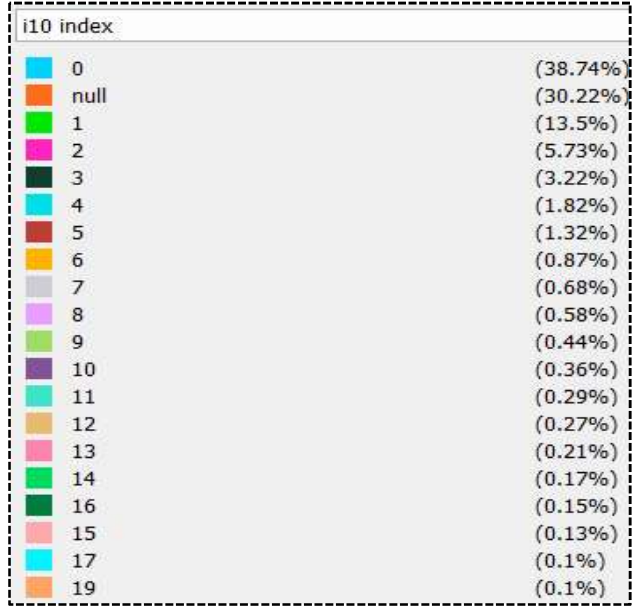
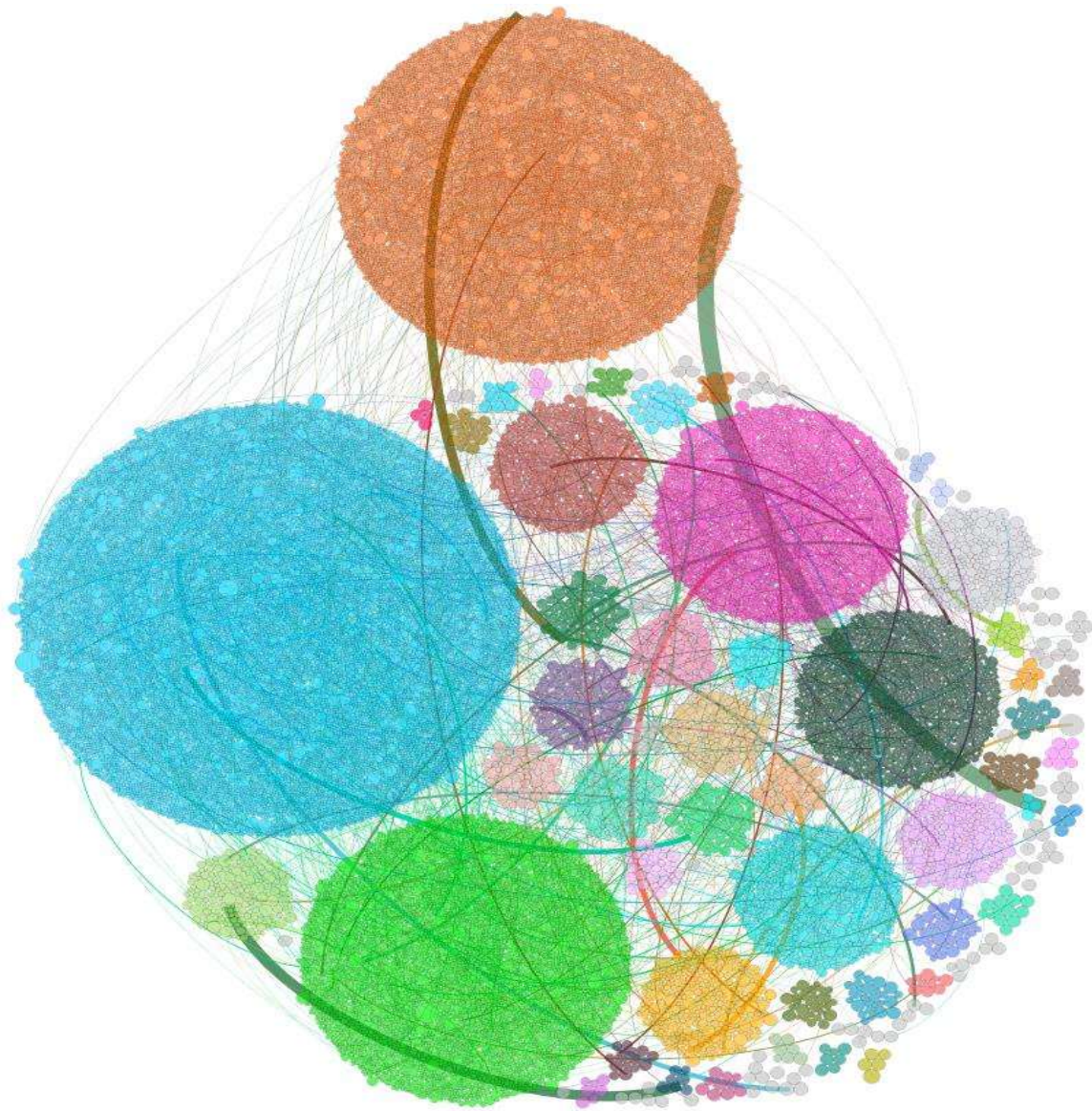


Figure 5.33 ISN Performance in term of i10-index.

5.3.5.2 IUN Performance Evaluation

In the previous sections, the evaluation of the ISN network was performed based on the authors' level. In this section, the evaluation is performed according to the university level.

A. Collaboration Among Universities

One of the strengths of complex networks is providing researchers with distinguished visualizations that enable a more accurate analysis of data. Therefore, a different visualization is performed in order to reveal the collaboration among the Iraqi universities. Figure 5.34 demonstrates each university as a bar and the length of each bar represents the size of a university, the longer the bars, the bigger the university. The edges among that connect the bars depict authors from a university who have co-authored papers with authors from other universities. However, the visualization needs zooming in order to clearly show the collaboration. More close views are shown in Figures 5.35 and 5.36, respectively which show the edges more clearly.

A microscopic view of the collaboration among the Iraqi universities is shown in Figure 5.37. This visualization demonstrates the nodes and edges more clearly. The larger nodes in the visualization correspond to authors with the highest frequency of paper authorship, providing a clearer representation of collaboration among Iraqi universities. On the other hand, the weight of the edges represents the frequency of collaboration (the number of coauthored papers). Different colors reflect different universities. The key colors of the nodes are the same as the Figures used previously when visualizing the ISN network without international authors.

When this network is used for strategic planning of scientific research in a particular institute, the main dataset file of the network should be used with the appropriate visualization tool. This is because these tools provide more analysis and more advanced visual inspection of the network in terms of the nodes, edges, and labels.

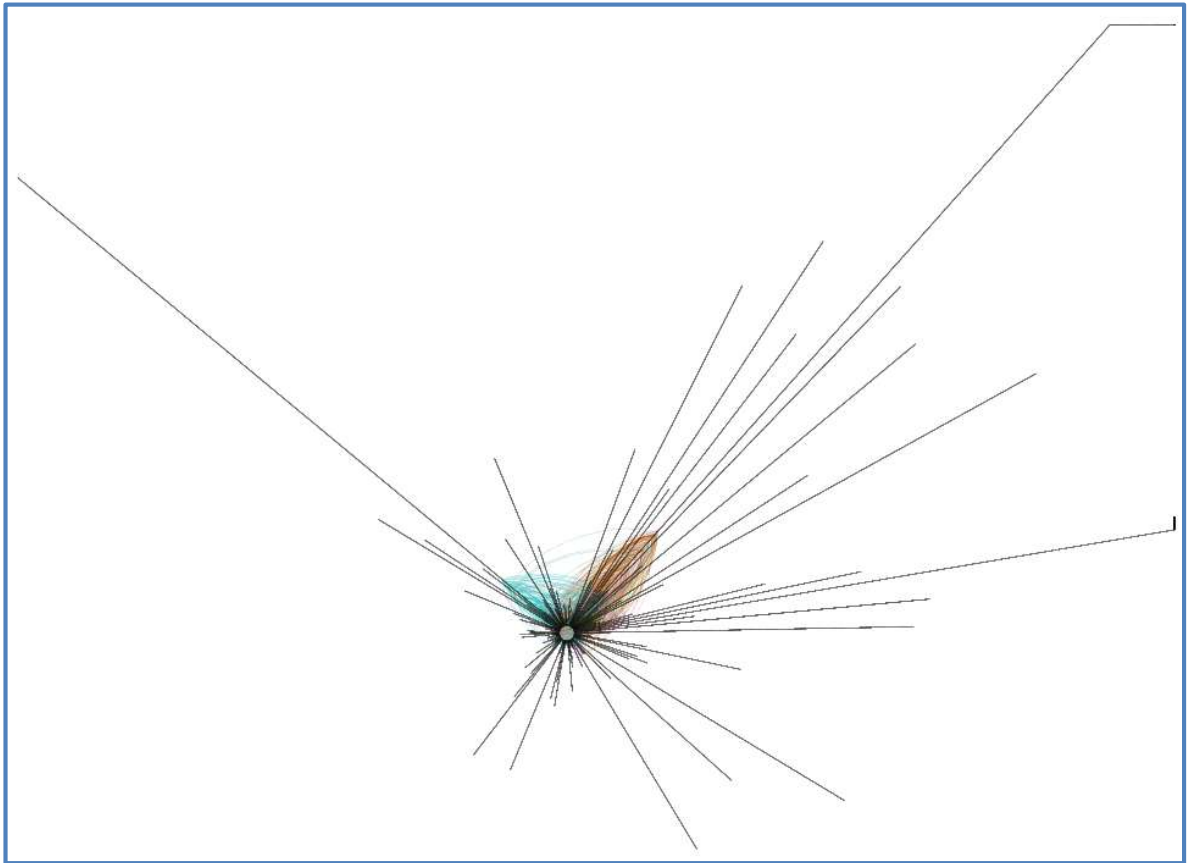


Figure 5 .34: Visualizing IUN as Bars.

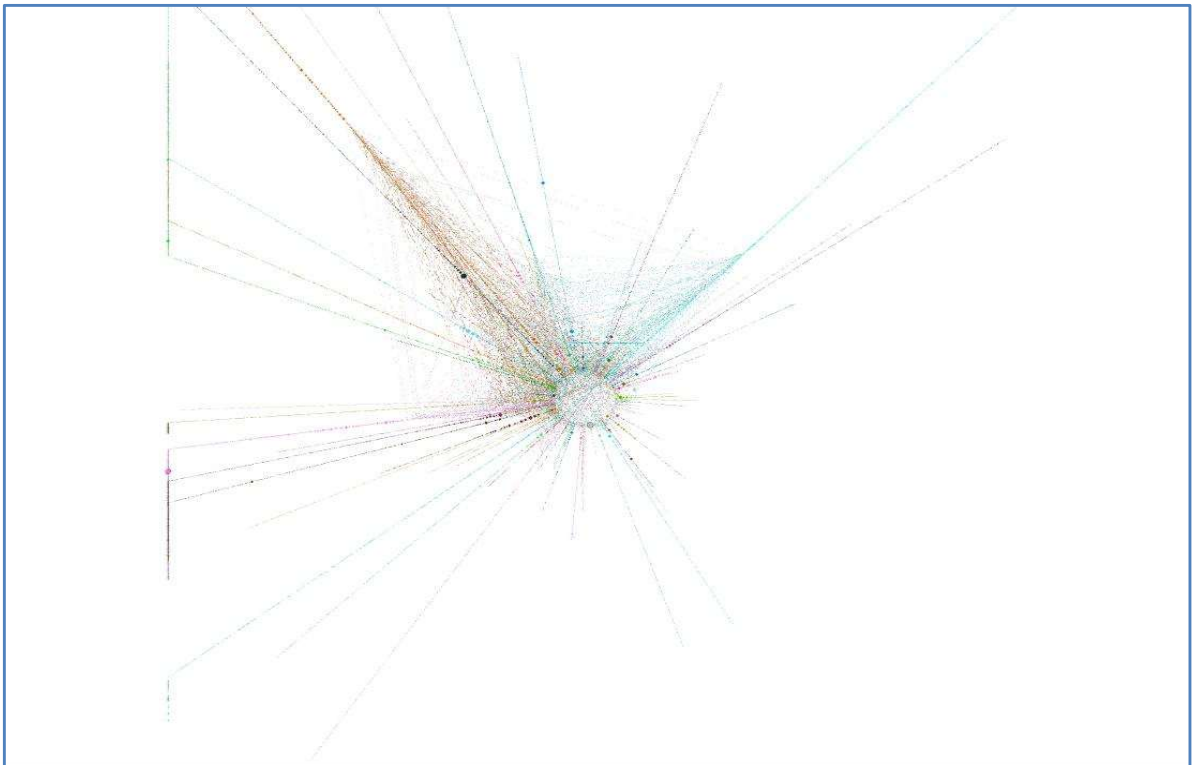


Figure 5.35 A Close View of the Collaboration Among the Iraqi Universities.

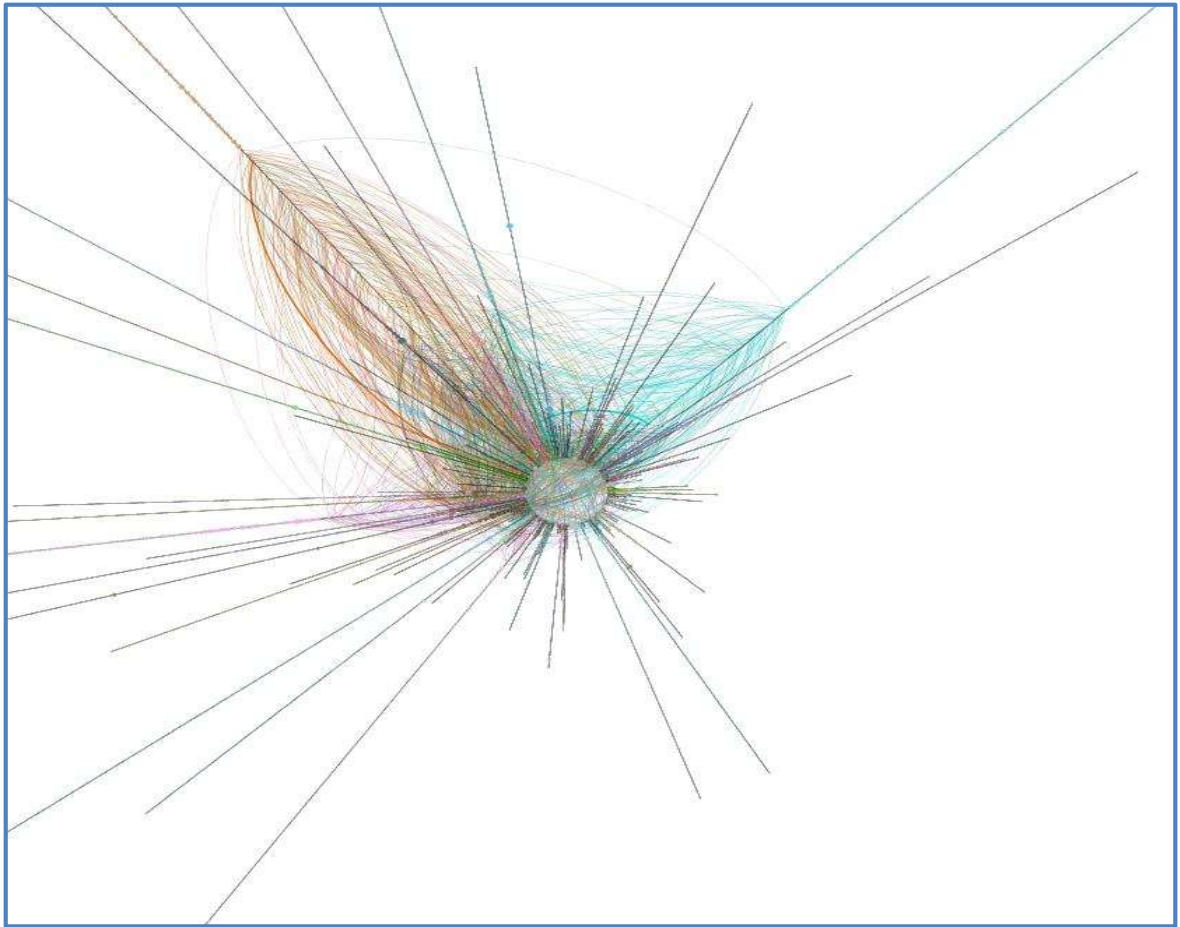


Figure 5.36 A Closer View of the Collaboration Among the Iraqi Universities



Figure 5.37 A Microscopic View to the Collaboration Among the Iraqi Universities.

B. Universities Network Analysis

The main characteristics of the Iraqi Universities Network (IUN) were presented in Table 5.5. The table shows that the number of Iraqi universities is 176, including public, private, and Kurdistan universities. The number of edges is 322, representing the relationships among the Iraqi universities. The diameter, the farthest distance between two nodes in the network, is 17, which is considered high compared to the network size. It should be mentioned that the distance of 17 is the number of steps to get between the farthest two universities in Iraq, which is significant, this means that there is a lack of collaboration among the Iraqi universities. In this case, a diameter of 17 suggests that there are relatively long paths between the farthest pairs of nodes in the network. This indicates that some parts of the network may be less directly connected, contributing to a higher average path length.

Table 5.5: Characteristics of the IUN Network.

# of Nodes	# of Edges	Diameter	Density	Average Clustering Coefficient	Average Shortest Path length
176	322	17	0.021	0.249	3.275

Also, the density of the edges in the network is too low compared to the size of the network, which also reflects a common tendency for collaboration between Iraqi universities. On the other hand, the average clustering coefficient reflects the tendency of nodes to cluster together. This means high levels of average clustering coefficient represent a high level of collaboration. The value of the average clustering coefficient should be between 0 and 1, and according to Table 5.5, it is relatively low, confirming the previous findings. Also, the average path length of 3.275 ensures the results that relatively low average path length (close to 1) indicates efficient communication and connectivity within the network. In this case, a value of 3.275 suggests a moderate distance between nodes on average.

As mentioned before, complex networks concepts are considered powerful tools that enable exploring facts in data that cannot be explored using traditional analysis. For this specific reason, a Geo Layout and Map of Countries are used

in the following visualizations. Figure 5.38 shows the visualization of the network, in which the real coordinates (based on Google Maps) of each university are projected on the Iraqi map to show their exact real locations and provide a comprehensive view of the Iraqi universities in terms of scientific collaboration. In this Figure, each university has a different color, and the size of the nodes represents the number of papers at the university (larger nodes reflect a large number of papers). The edges represent the relations between universities; thicker edges mean a high level of collaboration in authoring papers between two universities.

An important observation is extracted: cooperation between universities depends on their geographical distances. It turns out that the universities in the northern areas of Iraq have more collaboration between them. Similarly, the universities in the middle regions of Iraq have more collaborations than other sites in Iraq. This phenomenon can be interpreted as an indicator of a problem in the collaboration pattern between the Iraqi universities related to the geographical area.

Figure 5.39 depicts a power-law distribution of degrees, which means that there were a few universities in Iraq that have a high frequency of collaboration (co-authoring papers) with other universities, with majority of universities that have few collaborations. This finding tells us there is a gap between the senior universities and other universities, which should be addressed in the community of Iraqi universities. This gap can be filled by distributing influential authors to the universities that are considered less active in terms of collaboration and encouraging authors to collaborate with authors from other universities and consider it as a bonus when it comes to the authors' annual evaluation or scientific promotion. The collaboration is important because when different authors coauthor a paper, the paper itself will be produced from different points of view and from different scientific cultures and mentalities. Therefore, the level of collaboration between universities is considered an international indicator of the quality of the research or the university.

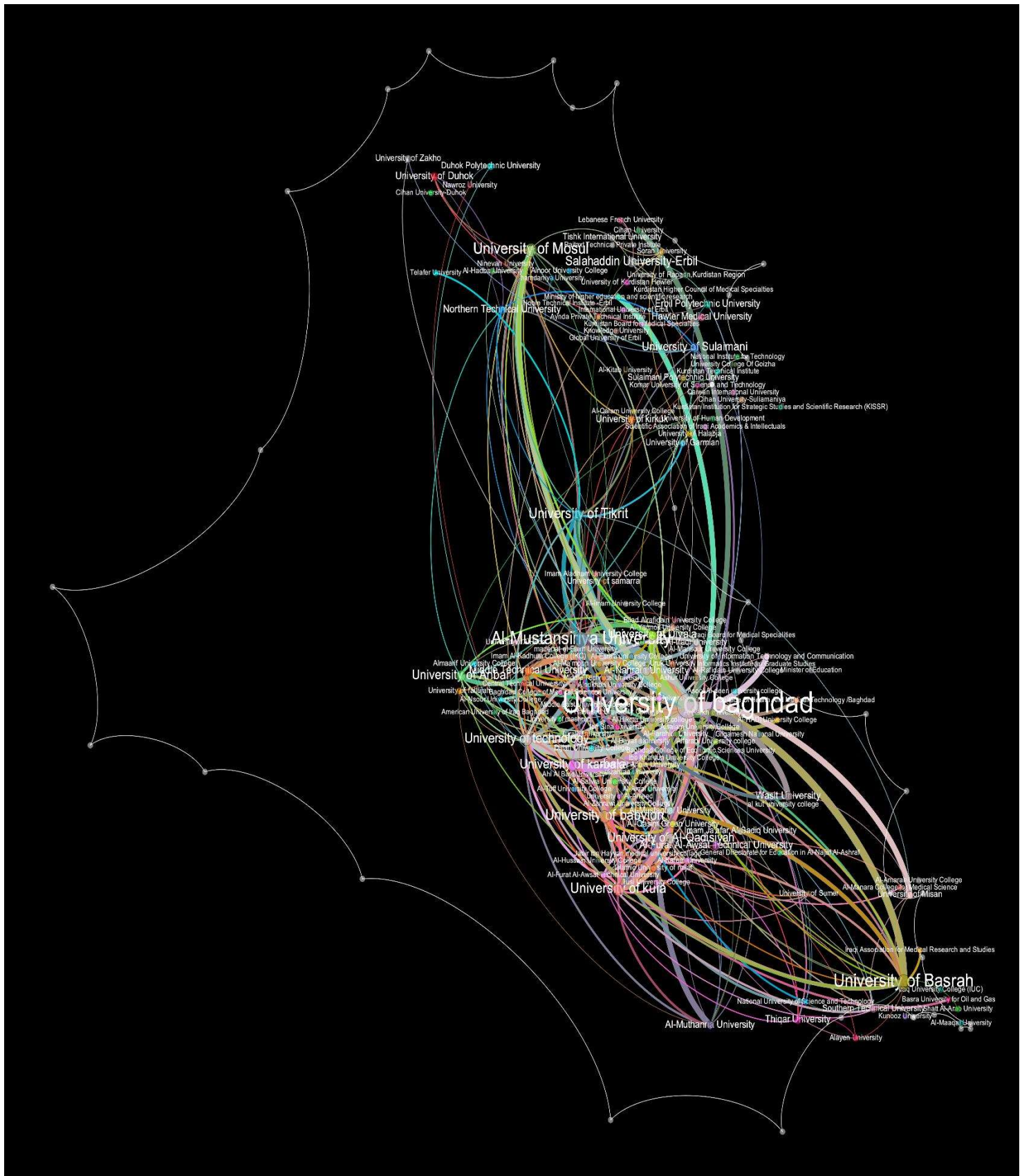


Figure 5.38 Geo Layout Visualization of the Iraqi Universities Alongside their Collaborations with other Iraqi Universities

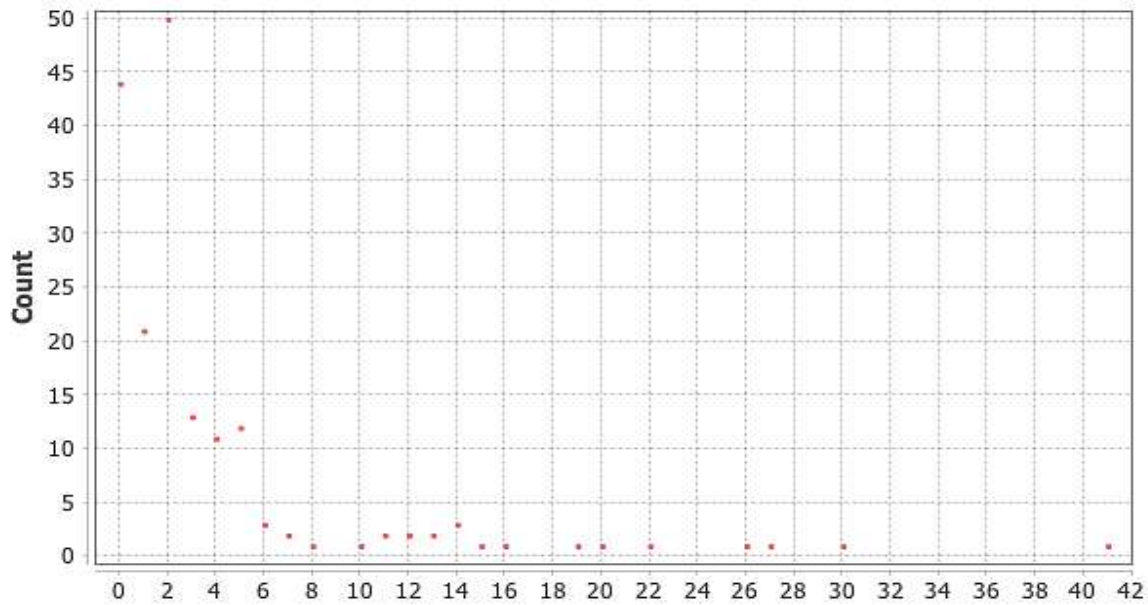


Figure 5.39 Degree Distribution

The betweenness centrality, on the other hand, shows that a few authors have played an influential role in the Iraqi research communities, while the majority have not (see Figure 5.40). The closeness centrality of the universities shows values around 0.5, which is acceptable (see Figure 5.41). The clustering coefficient of the universities shows that the majority of the universities, however, do not have a strong tendency to collaborate with other universities (see Figure 5.42). This problem can be addressed using strategies such as the one mentioned in the previous paragraph.

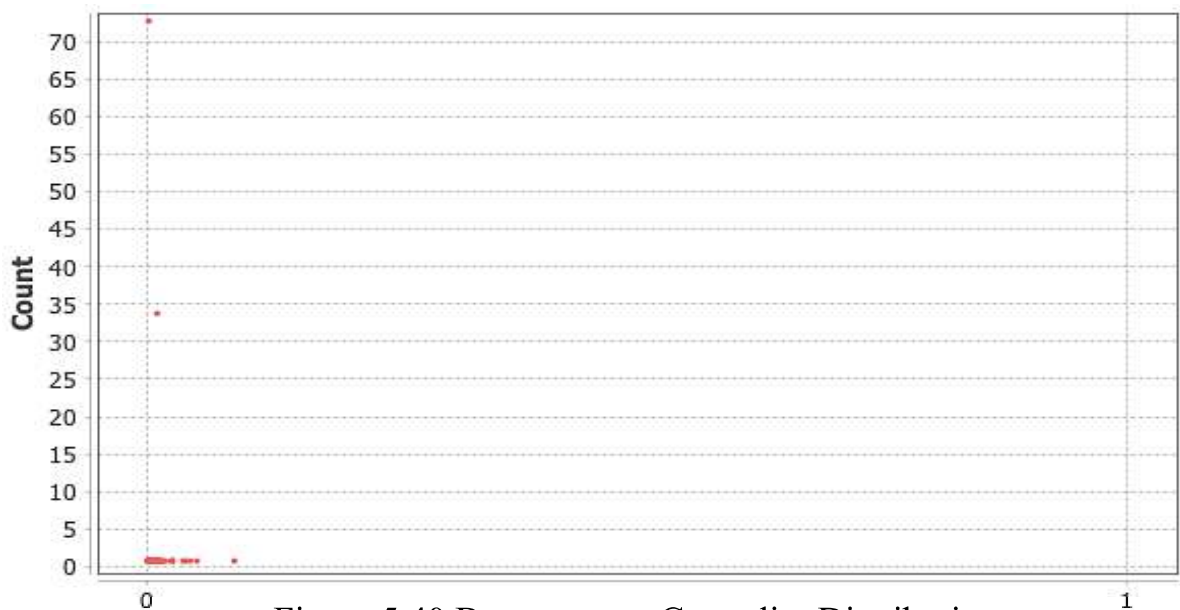


Figure 5.40 Betweenness Centrality Distribution

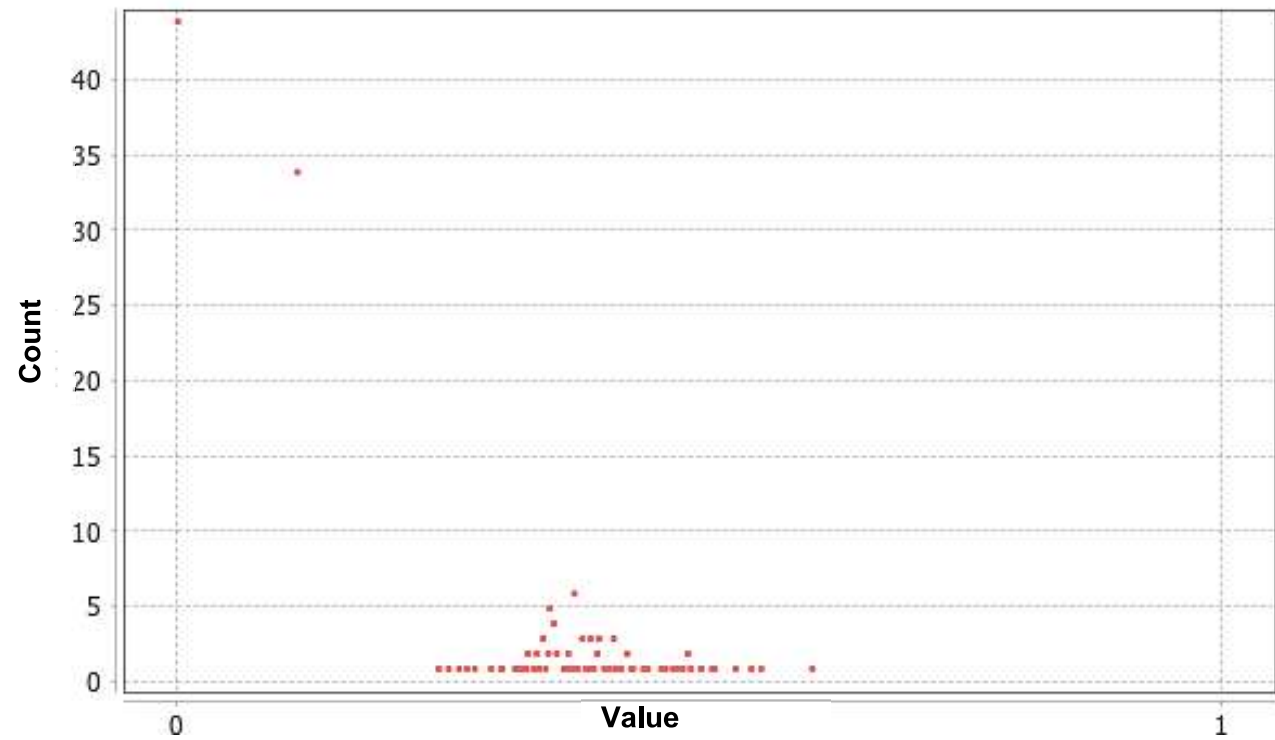


Figure 5.41 Closeness Centrality Distribution

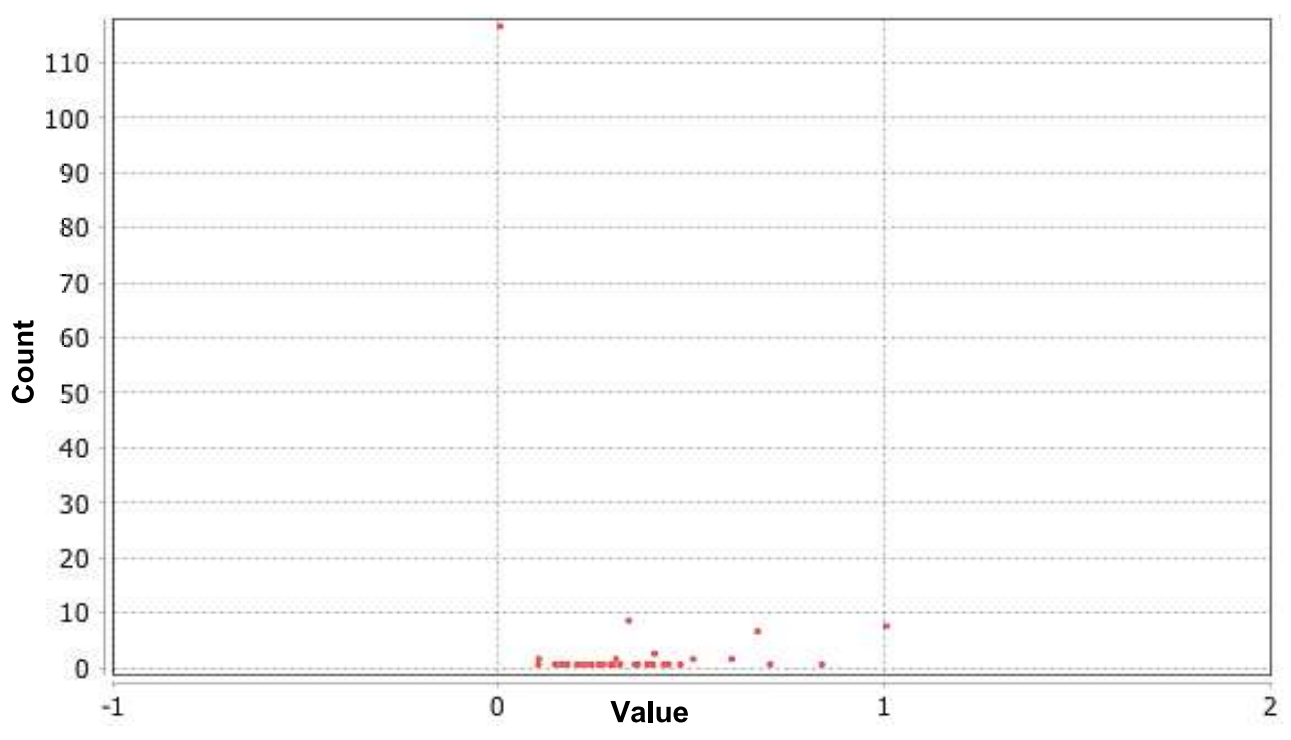


Figure 5.42 Clustering Coefficient Distribution

C. Edges Filtering Technique

The next step in the analysis of the IUN is to involve a filtering technique that enables in depth investigation of the collaboration, and finding out which universities are considered the most influential universities in terms of collaboration. The edges of IUN network were reduced by 25%, this means 25% of the least significant edges are removed from the network. In other words, the weakest 25% of edges are neglected from the network. Figure 5.43 visualizes the network and shows the collaboration after the edge reduction process (edge filtering). Additionally, Figure 5.44 show the visualization after removing the weakest 50% edges, and Figure 5.45 shows the collaboration after removing the weakest 75% of edges. These visualizations are interesting, as they show the actual performance of the Iraqi universities in terms of collaboration in coauthoring papers.

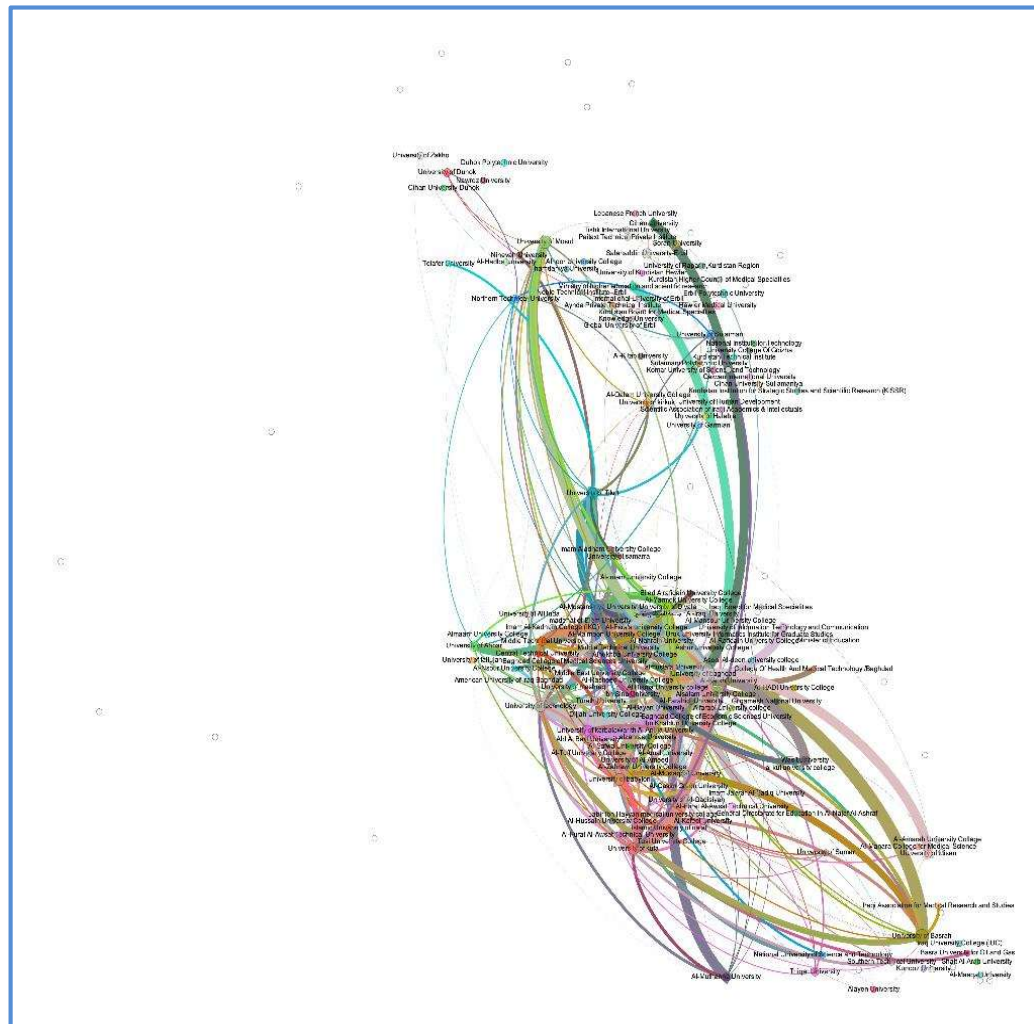


Figure 5.43 IUN Visualization After Removing the Weakest 25% Edges

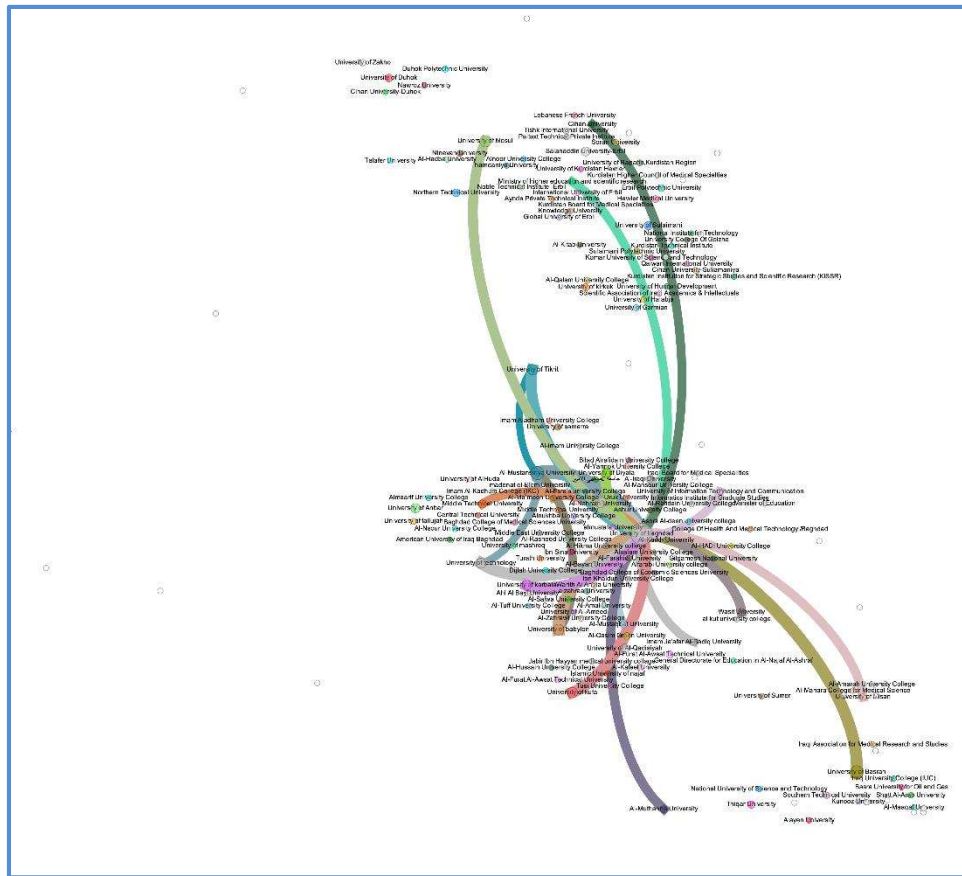


Figure 5.44 IUN Visualization After Removing the Weakest 50% Edges

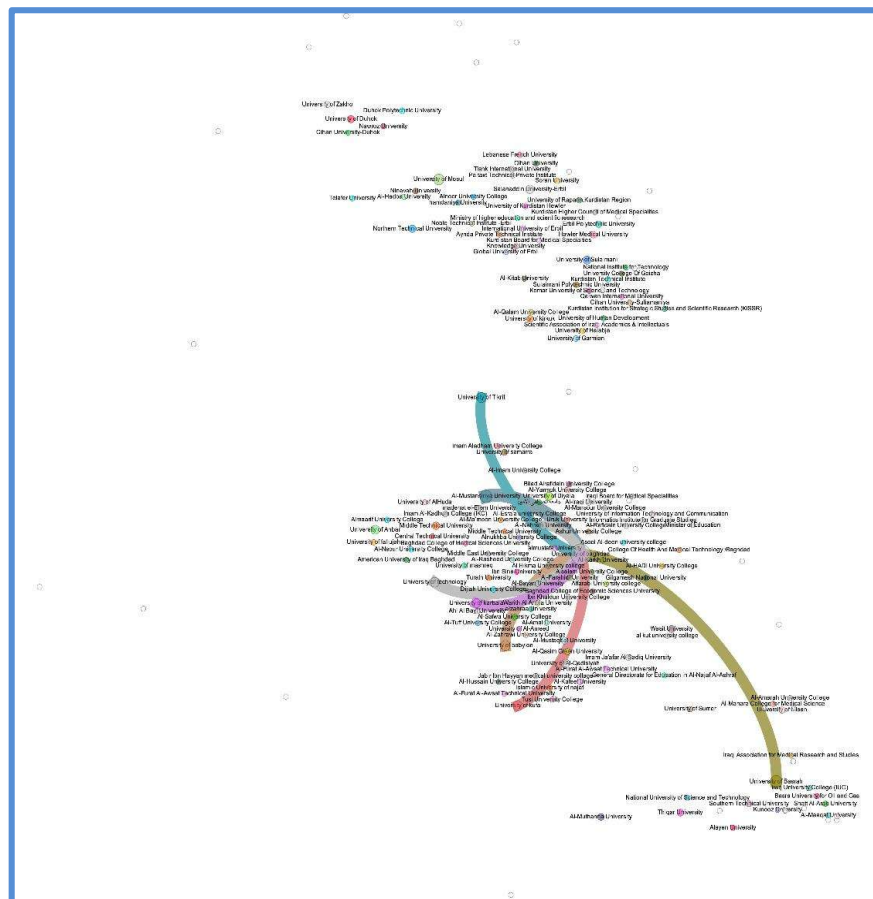


Figure 5.45 IUN Visualization After Removing the Weakest 75% Edges.

Figure 5.46 demonstrates the visualization of the IUN network with the top 25% collaborations. The visualization shows that the University of Baghdad, University of Tikrit, Al Mustansiriya University, University of Technology, University of Kerbala, University of Babylon, University of Kufa, University of Al-Nahrain and University of Basrah have top 25% of the collaborations among the Iraqi universities.

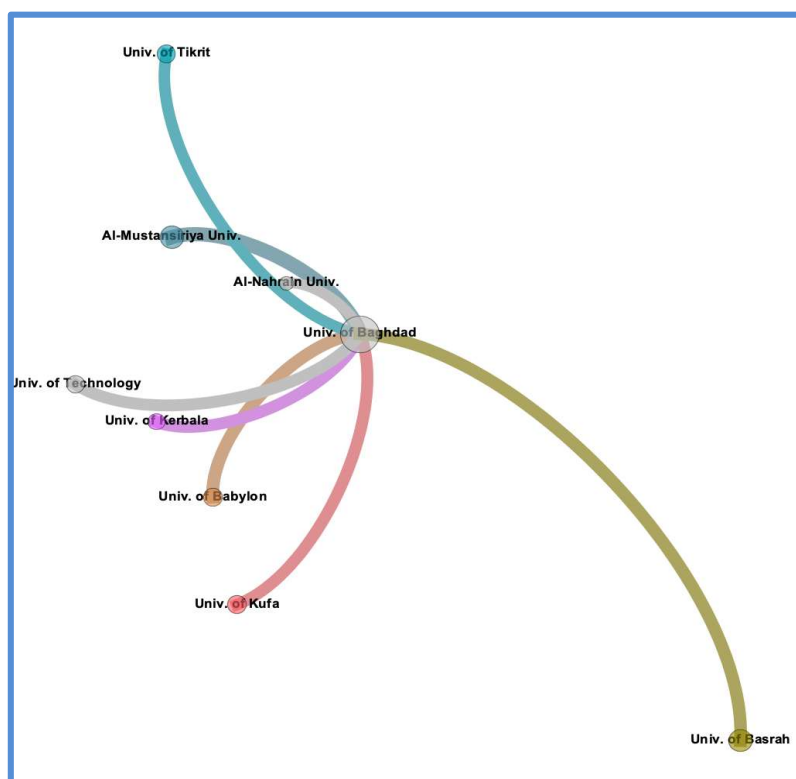


Figure 5.46 IUN Visualization of the Top 25% Collaboration

D. Iraqi Research Communities

In addition to the previous findings, the modularity of the communities of Iraqi universities is also investigated. The modularity of a network reflects the strength of the structure of the communities. Figure 5.47 demonstrates the level of modularity of all the communities of the authors in the Iraqi universities. Findings show that there are a total of 50 university communities in Iraq with a maximum modularity level of 0.256, which reflects weak modularity. The processing time in performing the modularity was 0.747 sec. This kind of modularity represents the standard community detection in Cytoscape software.

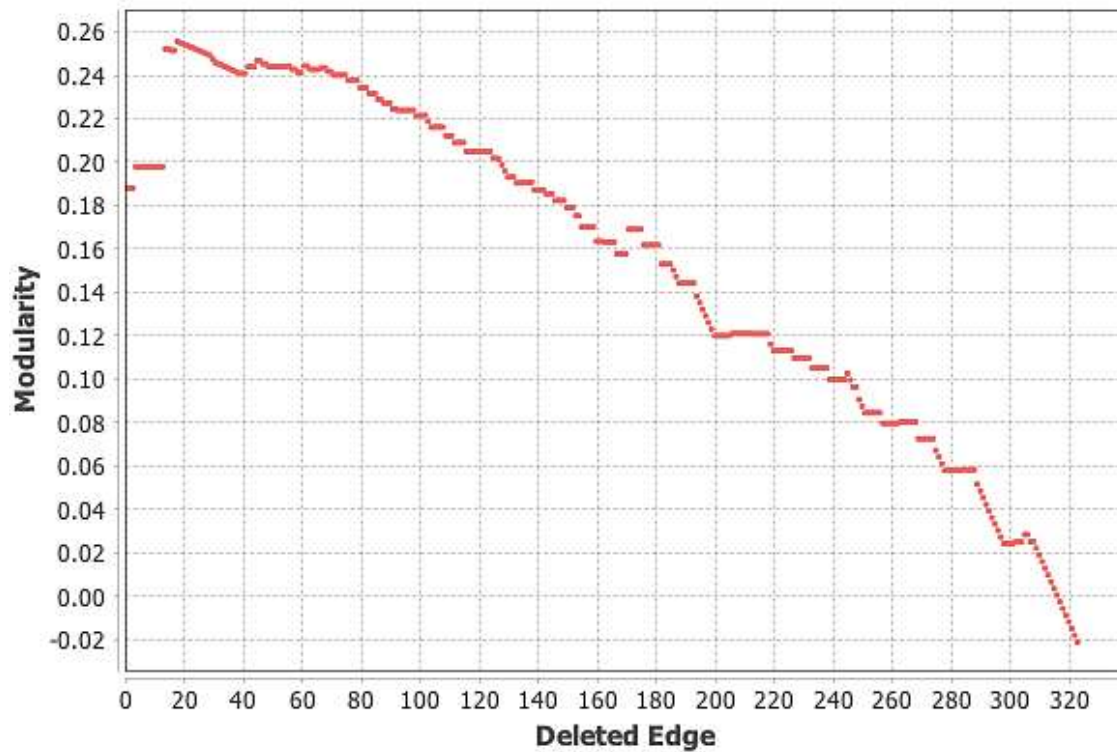


Figure 5.47 Distribution of Modularity in IUN Communities.

The number of communities in the IUN network were detected using two additional algorithms, namely Louvain and Leiden. These algorithms seek to identify cohesive groups of nodes within a network, termed communities, by optimizing a crucial metric known as modularity. Modularity measures the strength of the division of nodes into communities, favoring dense connections within communities and sparse connections between them. The Louvain algorithm operates iteratively, starting with each node in its own community and proceeding to optimize modularity by iteratively reassigning nodes to communities based on local structures and percolation patterns. The incorporation of CPM in Louvain algorithm enhances the algorithm's sensitivity to local structures, contributing to a nuanced exploration of potential community formations.

The Leiden algorithm, apart from its refined optimization strategy, introduces a meticulous iterative process to enhance the stability of community assignments. Operating with the goal of maximizing modularity, the algorithm iteratively refines community assignments, evaluating and adjusting them for greater stability. This method contributes to a more robust and reliable identification of communities within the network. The primary benefit lies in

the algorithm's ability to produce well-connected and internally cohesive communities, offering a clearer understanding of the intricate organization of complex networks.

Both algorithms were carried out using the collected dataset, with the application of identical settings and parameter values. In Table 5.6, the implementation details of the two algorithms were presented under the same settings, indicating that the number of communities formed by both algorithms is 46. This means Iraqi universities have 46 communities and these communities, following the Geo Layout visualization, depend on the geographical areas. It is noteworthy, the implementing of the aforementioned two algorithms used two different parameters for quality function: Modularity, which is the default parameter that is commonly used in these algorithms, and Constant Potts Model (CPM), which is used to overcome the limitations in the resolution.

Table 5.6: Community Detection Using Louvain and Leiden

Quality Function	LOUVAIN	LEIDEN	LOUVAIN	LEIDEN
	Modularity	Modularity	Constant Potts Model (CPM)	Constant Potts Model (CPM)
Resolution	0.01	0.01	0.05	0.05
Number of Iterations	100	100	200	200
Number of Restarts	2	2	5	5
Random Seed	Random	Random	Random	Random
Quality	99%	99%	99.7%	99.8%
Number of Communities	46	46	47	47

5.3.6 Academic Capital

In the previous chapter, the calculations of the academic capital were described in detail. The implementation and the results of the academic capital for authors and universities are demonstrated in this section. Figure 5.48 shows the distribution of authors' academic capital and how it varies from one author to another. It is clear that some authors have distinguished academic capital. In

this thesis point of view, the proposed academic capital reflects the actual performance of an author in a community of authors, as in its calculations, it takes into consideration many different dimensions of each author's performance, including the regular indicators (e.g., number of papers, citations, etc.) and other centrality indicators such as betweenness, closeness, etc. This combination enriches the academic capital with strength that cannot be found in any other metrics in the literature.

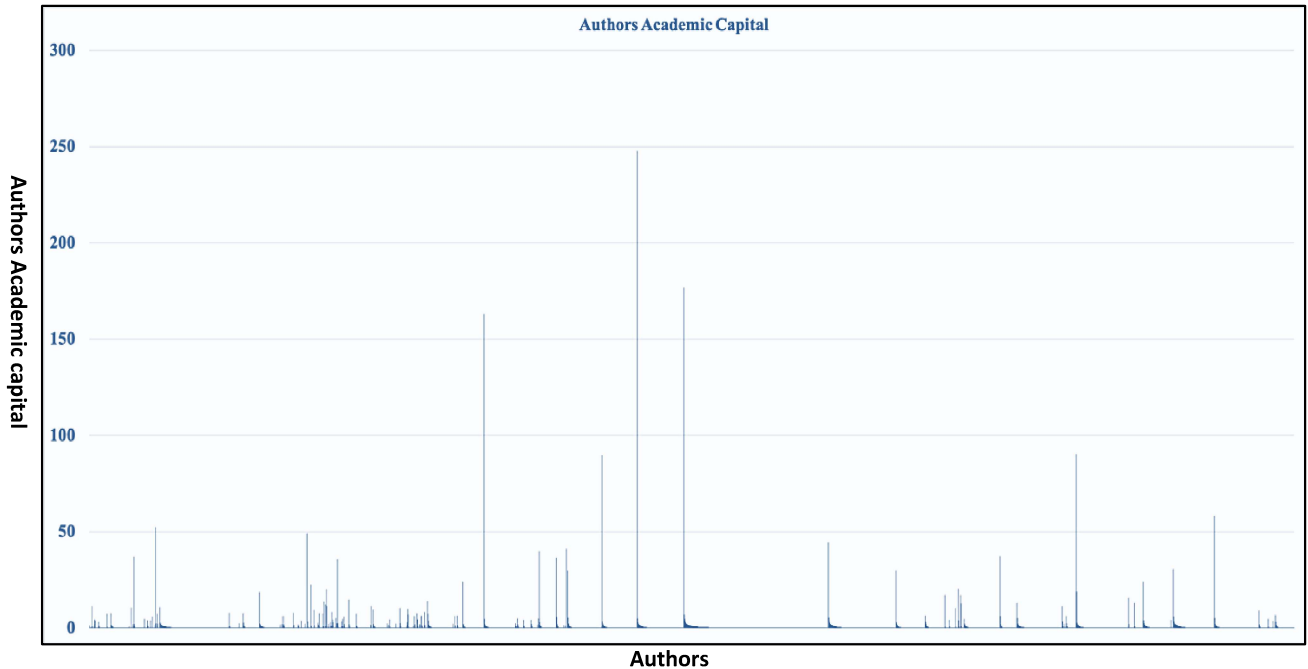


Figure 5.48 Authors' Academic Capital Distributions.

The academic capital of all authors in the Iraqi universities was calculated. Table 5.7 show the top 10 Iraqi authors with the highest academic capital along with their affiliations and specializations.

Table 5.7: Top 10 Iraqi Authors in Term of Academic Capital

	Author ID according to the dataset used	Author Affiliation	Interest According to Google Scholar	Academic Capital
1	800001	University of Babylon	Computer Sciences	247.7349968
2	800002	University of Baghdad	Sport Science	176.7975263
3	839287	Salahaddin University	Epidemiology	163.0491156
4	800003	University of Babylon	Physics	153.2192339
5	800004	University of Baghdad	Public Health	116.0427798
6	800006	University of Baghdad	Management	115.2164285
7	800007	University of Mosul	Mathematics	90.4304766
8	800008	University of Anbar	Arts	89.7820098
9	800013	University of Tikrit	Agricultural Science	58.3814758
10	800010	University of Tikrit	Genotoxicology	52.7828143

The academic capital of each university, on the other hand, was calculated by averaging the academic capital values of the authors of a university. Figure 5.49 demonstrates the performance of the academic capital of all the universities in Iraq. The figure shows that the University of Baghdad has the highest academic capital followed by Al-Mustansiriya, Basrah, Mosul, Babylon, and so on.

In this thesis, the use of academic capital as a rank for the universities was proposed. This means that the rank of a university is mainly based on its academic capital. For the purpose of evaluating the proposed rank, benchmarking was performed. The benchmarking was based on the popular local and international ranks such as the evaluation of Times, Webometrics, and QS. Table 5.8 presents the ranks of all the Iraqi universities along with the other ranks (ministry rank, webometrics, and QS). The table shows that the proposed rank is close to the other international ranks. However, the proposed rank is based on parameters that are of interest to the officials in Iraq as these parameters deeply investigate Google Scholar indicators as well as network centralities parameters, which are more comprehensive and accurate when evaluating universities.

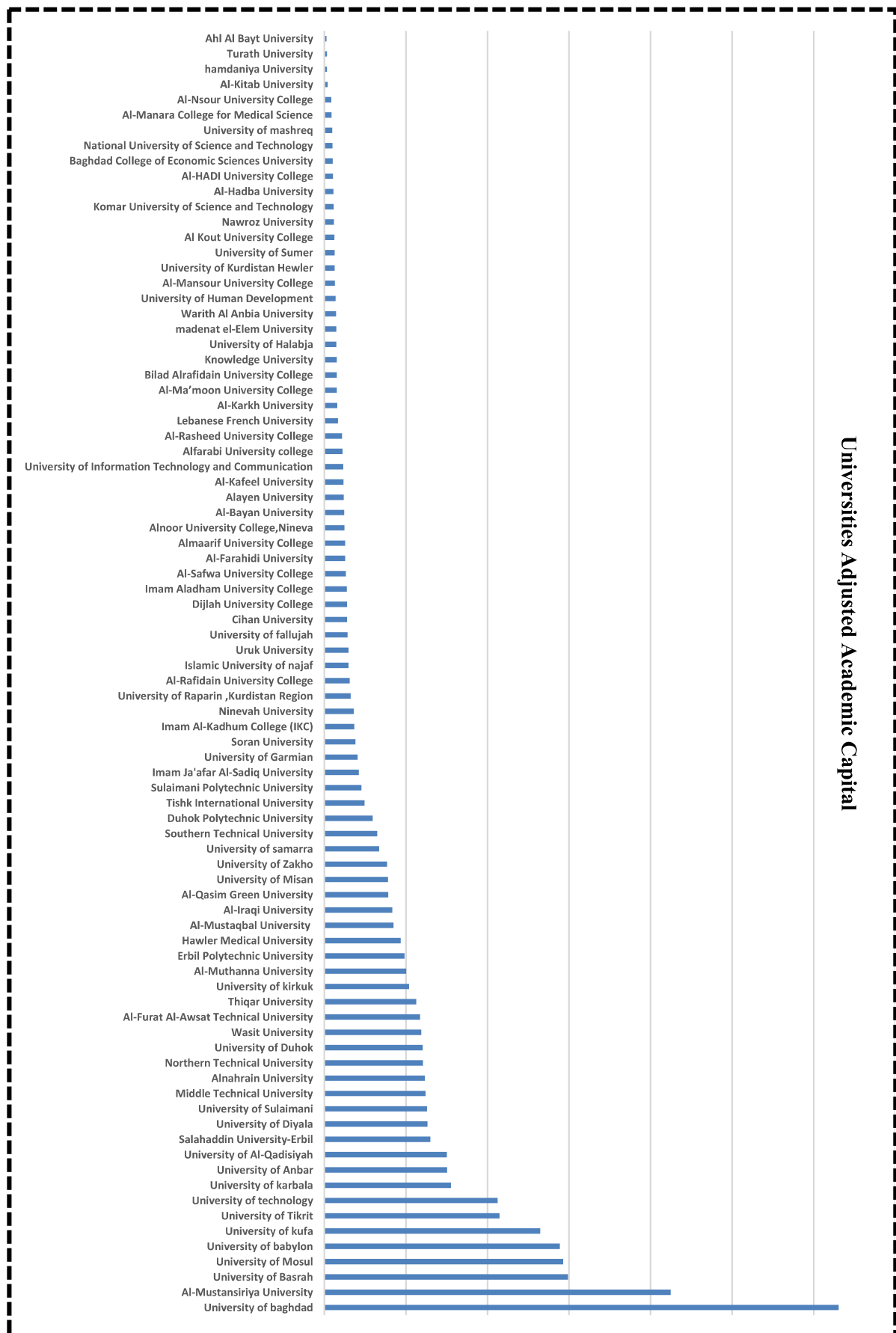


Figure 5.49 Academic Capital of the Iraqi Universities

Table 5.8 Benchmarking the Proposed Academic Capital-Based Rank with Other Local and International Ranks

University	Proposed Rank	Times Rank	Webometrics Rank	QS Rank
University of Baghdad	1	7	1	1
Al-Mustansiriya University	2	13	3	2
University of Basrah	3	8	22	4
University of Mosul	4	12	5	8
University of Babylon	5	4	8	3
University of Kufa	6	10	6	5
University of Tikrit	7	59	11	59
University of Technology	8	1	7	14
University of Kerbala	9	9	17	60
University of Anbar	10	6	9	53
University of Al-Qadisiyah	11	3	4	15
Salahaddin University-Erbil	12	55	15	52
University of Diyala	13	5	2	19
University of Sulaimani	14	/	10	9
Middle Technical University	15	11	27	45
Alnahrain University	16	2	14	6
Northern Technical University	17	53	44	46
University of Duhok	18	40	18	7
Wasit University	19	63	20	/
Al-Furat Al-Awsat Technical University	20	18	13	48
ThiQar University	21	58	23	21
University of Kirkuk	22	48	45	16
Al-Muthanna University	23	26	33	24
Erbil Polytechnic University	24	/	40	/
Hawler Medical University	25	42	21	13
Al-Mustaqbal University	26	25	26	39
Al-Iraqia University	27	/	46	10
Al-Qasim Green University	28	30	37	/
University of Misan	29	51	35	/
University of Zakho	30	/	16	18

University of Samarra	31	/	53	/
Southern Technical University	32	56	41	55
Duhok Polytechnic University	33	/	43	/
Tishk International University	34	60	12	29
Sulaimani Polytechnic University	35	/	30	25
Imam Ja'afar Al-Sadiq University	36	/	79	44
University of Garmian	37	41	24	/
Soran University	38	/	19	57
Imam Al-Kadhum College (IKC)	39	45	80	/
Ninevah University	40	52	57	27
University of Raparin, Kurdistan Region	41	/	38	/
Al-Rafidain University College	42	31	54	/
Islamic University of Najaf	43	46	34	38
Uruk University	44	61	99	/
University of Fallujah	45	/	58	26
Cihan University	46	38	28	/
Dijlah University College	47	/	47	/
Imam Aladham University College	48	/	98	11
Al-Safwa University College	49	32	65	/
Al-Farahidi University	50	/	74	/
Almaarif University College	51	23	52	40
Alnoor University	52	28	72	/
Al-Bayan University	53	15	68	/
Alayen University	54	14	49	34
Al-Kafeel University	55	20	60	47
University of Information Technology and Communication	56	/	39	28
Alfarabi University college	57	17	81	43
Al-Rasheed University College	58	/	63	/
Lebanese French University	59	/	31	17
Al-Karkh University	60	21	59	/
Al-Ma'moon University College	61	/	67	/
Bilad Alrafidain University College	62	36	56	/

Knowledge University	63	49	42	30
University of Halabja	64	/	71	36
madenat el-Elem University	65	/	77	/
Warith Al Anbia University	66	62	73	56
University of Human Development	67	44	29	/
Al-Mansour University College	68	/	50	/
University of Kurdistan Hewler	69	/	32	/
University of Sumer	70	/	62	23
Al Kout University College	71	/	85	32
Nawroz University	72	/	51	20
Komar University of Science and Technology	73	/	25	22
Al-Hadba University	74	/	111	/
Al-HADI University College	75	/	94	/
Baghdad College of Economic Sciences University	76	/	61	/
National University of Science and Technology	77	/	93	/
University of Mashreq	78	/	/	/
Al-Manara College for Medical Science	79	24	88	/
Al-Nisour University College	80	27	87	41
Al-Kitab University	81	22	66	42
Al-Hamdaniya University	82	19	83	/
Turath University	83	34	90	/
Ahl Al Bayt University	84	/	64	51

CHAPTER SIX

Conclusions, and Future Works

6.1 Preface

This chapter concludes the thesis in terms of the main observations about the results. The chapter also provides researchers and developers with recommendations that can be of interest when working in such a research area. The future works and research directions of this thesis are presented in this chapter.

6.2 Conclusions

The main conclusions of this thesis can be summarized as follows:

1. The novel rank metric, which is termed as Academic Capital Distributed Network (ACDN), gives valuable and feasible results about different universities in Iraq. This is due to the proposed metrics which dynamically consider multiple dimensions instead of solely focusing on the traditional dimensions of Google Scholar. As a result, this rank is also dynamic, adapting to changes in the scholarly landscape and research trends. An analysis of Academic Capital among authors in Iraqi universities reveals that the highest academic capital belongs to an author in the College of Computer Science in Babylon. Moreover, when considering the Academic Capital of Iraqi universities, the University of Baghdad emerges with the highest value, followed by Al-Mustansiriya University. This alignment with international ranking systems such as QS, Times, and Webometrics underscores the consistency and reliability of the findings, affirming the academic standing of these institutions within the global landscape.
2. The dynamic proposed design of the web crawler was based on specific programming techniques, which is able to efficiently extract biography information about authors, universities, and research items from Google Scholar.
3. The adoption of MongoDB Atlas significantly facilitates data management capabilities through robust infrastructure, which in turn allows efficient dynamic storage, organization, and retrievals of large datasets. The usage

it as a distributed database demonstrates its efficiency in data management, ensuring the integrity and accessibility of researcher's data.

4. The implementation of a cloud-based cluster proved effective. The integration of Apache Spark with this cluster facilitates parallel processing of data sourced from MongoDB Atlas. Furthermore, leveraging Spark SQL operations enables data processing and extraction of information from the collected biography data.
5. The concept of parallel processing and virtualization were successfully achieved and utilized. These two concepts significantly contribute to improving resource utilization, scalability, flexibility, and cost-efficiency dynamically in the data centers and cloud computing environments.
6. Concepts inspired by complex networks were highly useful in proposed two network models, the IUN and ISN. Also, they were useful and efficient in providing visual representations of the data of the Iraqi authors and universities.
7. The analysis reveals that the average degree of the Iraqi Scholar Network (ISN) stands at 1.8, indicating that each Iraqi scholar has contributed to approximately two papers on average, a figure comparable to the international standard set by ACM. Moreover, the average shortest path length of IUN is 3.2, demonstrating a high level of scientific collaboration among Iraqi universities. However, the clustering coefficient of the IUN, at 0.249, suggests a lack of strong connections among Iraqi universities and a low tendency to cluster together.

6.3 Future Works

The future works and directions of this thesis can be stated as follows:

1. Involve more advanced data preprocessing techniques to improve data quality and consistency by implementing natural language processing (NLP) algorithms to allow deeper insights to be extracted from academic texts.
2. Utilize the concepts of machine learning within Spark aiming for more advanced data analysis and prediction tasks, such as graph databases for

network analysis or machine learning frameworks for predictive modeling.

3. Use serverless computing options that enable the dynamic allocation of resources based on workload demands. This approach can significantly reduce operational costs by efficiently utilizing resources as needed, thereby optimizing resource usage and minimizing expenses.
4. Scalable the scope of work to be global, that is, including universities of other countries.
5. Construct and develop a standardized rank for other platforms like (Scopus, research gate, academia).
6. Develop a specialized software tool to gather biographical data from diverse platforms such as WOS, Scoups, streamlining the information gathering process for academic research.

References

- Abdel-Fattah, M. A., Mohamed, W., & Abdelgaber, S. (2022). A Comprehensive Spark-Based Layer for Converting Relational Databases to NoSQL. *Big Data and Cognitive Computing*, 6(3), 71.
- Abuabdo, A., & Al-Sharif, Z. A. (2019). Virtualization vs. containerization: Towards a multithreaded performance evaluation approach. 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA),
- Afolabi, I. T., Ayo, A., & Odetunmbi, O. A. (2021). Academic collaboration recommendation for computer science researchers using social network analysis. *Wireless Personal Communications*, 121(1), 487-501.
- Agarwal, A., Rao, S. K. S., & Mahendra, B. (2020). Comprehensive review of virtualization tools. *Int. Res. J. Eng. Technol*, 7(6).
- Agreement, G. Distributed Real-time Architecture for Mixed Criticality Systems.
- Ahmadian, H., Obermaisser, R., & Perez, J. (2018). Distributed Real-Time Architecture for Mixed-Criticality Systems. CRC Press.
- Ahmed, N. (2022). Performance modelling, analysis and prediction of Spark jobs in Hadoop cluster: a thesis by publications presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science, School of Mathematical & Computational Sciences, Massey University, Auckland, New Zealand Massey University].
- Akilandeswari, P., Bettala, S. D., Alankritha, P., Srimathi, H., & Krithik Sudhan, D. (2019). Dynamic Data Auditing Using MongoDB in Cloud Platform. *International Conference on Computer Networks and Communication Technologies: ICCNCT 2018*,
- Al Husaeni, D. F., & Nandiyanto, A. B. D. (2022). Bibliometric using Vosviewer with Publish or Perish (using google scholar data): From step-by-step processing for users to the practical examples in the analysis of digital learning articles in pre and post Covid-19 pandemic. *ASEAN Journal of Science and Engineering*, 2(1), 19-46.
- Alba, F. G. (2022). Academic human capital and research performance Universidad de Cádiz].

- Alberto, P. (2021). Privacy Preserving Data Mining: a distributed approach to data anonymization Politecnico di Torino].
- Al-Kerboly, D. M. A. (2023). A Hybrid Integrated Model for Big Data Applications based on Web Mining Technique and Recommender Systems, Ph.D. Thesis, College of Computer Science and Information Technology, University of Anbar.
- Al-Khayat, M. S., Al-Othman, M. S., & Al-Hakeem, M. S. (2016). A proposed cloud computing model for Iraqi's engineering colleges and institutes. ZANCO Journal of Pure and Applied Sciences, 28(2), p1-5.
- Andersson, P. (2021). Developing a Python based web scraper: A study on the development of a web scraper for TimeEdit. In.
- Aung, T. T., & Nyunt, T. T. S. (2020). Community detection in scientific co-authorship networks using neo4j. 2020 IEEE Conference on Computer Applications (ICCA),
- Azhar, R. J. K., Nurhakim, L., & Putra, R. E. (2019). Implementasi Web Scraping Untuk Menampilkan Informasi Tayangan Film di Bioskop: Book My Show. Universitas Siliwangi, 1, 1-7.
- Balusamy, B., Kadry, S., & Gandomi, A. H. (2021). Big Data: Concepts, Technology, and Architecture. John Wiley & Sons.
- Berisha, B., Mëziu, E., & Shabani, I. (2022). Big data analytics in Cloud computing: an overview. Journal of Cloud Computing, 11(1), 24.
- Bernard, J., & Kintziger, T. (2021). MongoDB Code Smells: Defining, Classifying and Detecting Code Smells for MongoDB Interactions in Java Programs Master's thesis, Faculté d'informatique. Université de Namur, Belgium].
- Bitzer, E., & Matimbo, F. (2017). Cultivating African academic capital–Intersectional narratives of an African graduate and his PhD study supervisor. Innovations in Education and Teaching International, 54(6), 539-549.
- Broucke, S. V., & Baesens, B. (2018). Practical Web Scraping for Data Science: best practices and examples with Python. Apress.
- Burguillo, J. C. (2018). Complex Networks. 29, 35-56. https://doi.org/10.1007/978-3-319-69898-4_3
- Buyya, R., Calheiros, R. N., & Dastjerdi, A. V. (2016). Big data: principles and paradigms. Morgan Kaufmann.

- CARLE, V. (2020). Web Scraping using Machine Learning.
- Chanda, B. (2021). A Parallel Processing Technique for Filtering and Storing User Specified Data Carleton University].
- Chauhan, A. (2019). A review on various aspects of MongoDB databases. *Int. J. Eng. Res. Sci. Technol*, 8(5), 90-92.
- Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, 19, 171-209.
- Chi, D., Tang, C., & Yin, C. (2021). Design and Implementation of Hotel Big Data Analysis Platform Based on Hadoop and Spark. *Journal of Physics: Conference Series*,
- Choudhary, A., Verma, P. K., & Rai, P. (2021). A walkthrough of Amazon Elastic Compute Cloud (Amazon EC2): A Review. *International Journal for Research in Applied Science and Engineering Technology*, 9(11), 93-97.
- Chythanya, K. R., Sunil, G., Kumar, K. S., Korra, S. N., & Harshavardhan, A. (2019). Security and Safety in Amazon EC2 Service—A Research on EC2 Service AMIs. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN, 2278-3075.
- Dale, K. (2022). *Data Visualization with Python and JavaScript*. " O'Reilly Media, Inc."
- Delgado López-Cózar, E., Orduña-Malea, E., & Martín-Martín, A. (2019). Google Scholar as a data source for research assessment. *Springer handbook of science and technology indicators*, 95-127.
- Doncheva, N. T., Morris, J. H., Gorodkin, J., & Jensen, L. J. (2018). Cytoscape StringApp: network analysis and visualization of proteomics data. *Journal of proteome research*, 18(2), 623-632.
- Dorđević, B., Jovičić, I., Kraljević, N., & Timčenko, V. (2022). Comparison of type-2 hypervisor performance on the example of VirtualBox, VMware Workstation player and MS Hyper-V. *PROCEEDINGS IX ICETRAN*.
- Edward, S. G., & Sabharwal, N. (2015). *Practical MongoDB: Architecting, Developing, and Administering MongoDB*. Apress.
- Elmeiligy, M. A., Desouky, A. I. E., & Elghamrawy, S. M. (2021). An efficient parallel indexing structure for multi-dimensional big data using spark. *The Journal of Supercomputing*, 77(10), 11187-11214.

- Feng, W. (2018). Learning Apache Spark with Python Release v1.0.
- Fonseca, B. d. P. F. e., Sampaio, R. B., Fonseca, M. V. d. A., & Zicker, F. (2016). Co-authorship network analysis in health research: method and potential use. *Health research policy and systems*, 14, 1-10.
- Fujita, M. (2019). Analysis of Researchers Using Network Centralities of Co-authorship from the Academic Literature Database. *Evolutionary Computing and Artificial Intelligence: Essays Dedicated to Takao Terano on the Occasion of His Retirement 2*, 57-63.
- Fujita, M., Inoue, H., & Terano, T. (2017). Searching promising researchers through network centrality measures of co-author networks of technical papers. *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*,
- Fujita, M., Inoue, H., & Terano, T. (2021). Analyzing promising researchers using network centralities of co-authorship networks from academic literature. *New Generation Computing*, 39, 181-197.
- Ganesh, D., Akhila, M. I., & Reddy, M. K. S. (2022). Potential Load-Based Virtualization Selection on Overloaded Cloud Hosts. *Journal of Algebraic Statistics*, 13(2), 2053-2059.
- Gupta, S. (2018). Performance Evaluation of Unstructured PBRA for Bigdata with Cassandra and MongoDB in Cloud. *International Journal of Cloud Applications and Computing (IJCAC)*, 8(3), 48-59.
- Gusenbauer, M. (2019). Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics*, 118(1), 177-214.
- Halas, F. (2017). Performance Measurement of Stream Data Processing in Apache Spark. In: Faculty of Informatics, Masaryk University, Brno, Czech Republic.
- Hamstra, M., & Zaharia, M. (2013). Learning Spark: lightning-fast big data analytics. O'Reilly & Associates.
- Higaki, A., Uetani, T., Ikeda, S., & Yamaguchi, O. (2020). Co-authorship network analysis in cardiovascular research utilizing machine learning (2009–2019). *International Journal of Medical Informatics*, 143, 104274.
- Hill, R., Trovati, M., Liu, L., Anjum, A., & Zhu, S. Y. (2015). Big-Data analytics and cloud computing: Theory, algorithms and applications. In: 921y4.

- Hows, D., Membrey, P., & Plugge, E. (2014). *MongoDB Basics*. Springer.
- Hwang, K., Dongarra, J., & Fox, G. C. (2013). *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan kaufmann.
- Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., & Nguifo, E. M. (2018). An experimental survey on big data frameworks. *Future Generation Computer Systems*, 86, 546-564.
- Iskandar, A., Aziz, F., Dewi, R. D. C., Rusli, R., & Ahmar, A. S. (2020). Co-authorship visualization of research on COVID-19 from Web of Science data using bibliometric analysis. *Library Philosophy and Practise (e-journal)*, 4528, 1-9.
- Jaiswal, A., Dwivedi, V. K., & Yadav, O. P. (2020). Big data and its analyzing tools: a perspective. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS),
- Johansson, P. (2020). Dynamic Co-authorship Network Analysis with Applications to Survey Metadata. In.
- Johnson, A., Mercado, C., & Khan, S. (2022). Improving Diversity in Journalistic Sources with Computer Vision.
- Karataş, G., & Akbulut, A. (2018). Survey on access control mechanisms in cloud computing. *Journal of Cyber Security and Mobility*, 7(3), 1-36.
- Kartun-Giles, A. P., & Bianconi, G. (2019). Beyond the clustering coefficient: A topological analysis of node neighbourhoods in complex networks. *Chaos, Solitons & Fractals*: X, 1, 100004. <https://doi.org/10.1016/j.csfx.2019.100004>
- Khalid, M., & Yousaf, M. M. (2021). A comparative analysis of big data frameworks: An adoption perspective. *Applied Sciences*, 11(22), 11033.
- Khalil, S., & Fakir, M. (2017). RCrawler: An R package for parallel web crawling and scraping. *SoftwareX*, 6, 98-106.
- Khder, M. A. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).
- Kholod, S.-V. (2021). Performance comparison for different types of databases.
- Kisseljov, A. (2023). Following ICT bachelor programme offerings (studyinfo. fi) using web scraping with Python. Master. Thesis, Information and Communication Technology, Turku University of Applied Sciences.

- Kronenberg, N. P., Levy, H. M., & Strecker, W. D. (1986). VAXcluster: A closely-coupled distributed system. *ACM Transactions on Computer Systems (TOCS)*, 4(2), 130-146.
- Krotov, V., Johnson, L., & Silva, L. (2020). Tutorial: Legality and ethics of web scraping.
- Kumar, P., & Kumar, R. (2019). Issues and Challenges of Load Balancing Techniques in Cloud Computing. *ACM Computing Surveys*, 51(6), 1-35. <https://doi.org/10.1145/3281010>
- Kumar, S., & Kumar, R. (2021). A study on different aspects of web mining and research issues. *IOP conference series: Materials Science and Engineering*,
- Kumari, K. A., Sadasivam, G. S., Dharani, D., & Niranjnamurthy, M. (2021). *Edge Computing: Fundamentals, Advances and Applications*.
- Lai, M., & Shi, L. (2023). How academics perceive their participation in applied projects: a case study of a third-tier university on the Chinese Mainland. *Cambridge Journal of Education*, 1-15.
- Le, P. (2022). Development of an eCommerce website for Ngoc's MaxiNutri Company.
- Lima, T. S., de Arruda, H. F., Silva, F. N., Comin, C. H., Amancio, D. R., & Costa, L. d. F. (2018). The dynamics of knowledge acquisition via self-learning in complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(8).
- Liu, J., Pacitti, E., & Valduriez, P. (2018). A survey of scheduling frameworks in big data systems. *International Journal of Cloud Computing*, 7(2), 103-128.
- Lula, P., Dospinescu, O., Homocianu, D., & Sireteanu, N.-A. (2021). An Advanced Analysis of Cloud Computing Concepts Based on the Computer Science Ontology. *Computers, Materials & Continua*, 66(3), 2425-2443. <https://doi.org/10.32604/cmc.2021.013771>
- Luo, Y., Card, D., & Jurafsky, D. (2020). Detecting stance in media on global warming. *arXiv preprint arXiv:2010.15149*.
- Malik, M. I., Wani, S. H., & Rashid, A. (2018). CLOUD COMPUTING-TECHNOLOGIES. *International Journal of Advanced Research in Computer Science*, 9(2).
- Martín-Martín, A., & López-Cózar, E. D. (2021). Large coverage fluctuations in Google Scholar: a case study. *arXiv preprint arXiv:2102.07571*.

- Martín-Martín, A., Orduna-Malea, E., & Delgado López-Cózar, E. (2018). A novel method for depicting academic disciplines through Google Scholar Citations: The case of Bibliometrics. *Scientometrics*, 114, 1251-1273.
- Martín-Martín, A., Orduña-Malea, E., Harzing, A.-W., & López-Cózar, E. D. (2017). Can we use Google Scholar to identify highly-cited documents? *Journal of informetrics*, 11(1), 152-163.
- Martín-Martín, A., Orduna-Malea, E., Thelwall, M., & López-Cózar, E. D. (2018). Google Scholar, Web of Science, and Scopus: A systematic comparison of citations in 252 subject categories. *Journal of informetrics*, 12(4), 1160-1177.
- Martín-Martín, A., Thelwall, M., Orduna-Malea, E., & Delgado López-Cózar, E. (2021). Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations. *Scientometrics*, 126(1), 871-906.
- Meesters, G. (2021). Synchronising distributed scraping. Open University of the Netherlands, Heerlen.
- Mishra, S. (2020). Social networks, social capital, social support and academic success in higher education: A systematic review with a special focus on 'underrepresented' students. *Educational Research Review*, 29, 100307.
- Mitchell, R. (2018). Web scraping with Python: Collecting more data from the modern web. " O'Reilly Media, Inc."
- Mittal, M., Balas, V. E., Goyal, L. M., & Kumar, R. (2019). Big data processing using spark in cloud. Springer.
- Orduña-Malea, E., Ayllón, J. M., Martín-Martín, A., & Delgado López-Cózar, E. (2015). Methods for estimating the size of Google Scholar. *Scientometrics*, 104, 931-949.
- Osipov, D. (2019). Development of a MongoDB-connected VR application.
- Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. *Future Generation Computer Systems*, 91, 620-633.
- Omar, H. K., & Jumaa, A. K. (2022). Distributed big data analysis using spark parallel data processing. *Bulletin of Electrical Engineering and Informatics*, 11(3), 1505-1515.
- Parlina, A., Ramli, K., & Murfi, H. (2020). Theme mapping and bibliometrics analysis of one decade of big data research in the scopus database. *Information*, 11(2), 69.

- Pessoa Junior, G. J., Dias, T. M., Silva, T. H., & Laender, A. H. (2020). On interdisciplinary collaborations in scientific coauthorship networks: the case of the Brazilian community. *Scientometrics*, 124, 2341-2360.
- Rahman, Tazyn & Jagannarayan, Nandini & Kannan, Aishwariya. (2021). *Advances in Management, Social Sciences and Technology* By Dr. Tazyn Rahman (2).
- Ramírez-Gallego, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Benítez, J. M., Alonso-Betanzos, A., & Herrera, F. (2017). An information theory-based feature selection framework for big data under apache spark. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1441-1453.
- Rattanavayakorn, P., & Premchaiswadi, W. (2015). Analysis of the social network miner (working together) of physicians. 2015 13th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering 2015),
- Rossetti, G., & Cazabet, R. (2018). Community discovery in dynamic networks: a survey. *ACM computing surveys (CSUR)*, 51(2), 1-37.
- Rovira, C., Codina, L., & Lopezosa, C. (2021). Language bias in the Google Scholar ranking algorithm. *Future Internet*, 13(2), 31.
- Royal, R. S. (2022). A Mixed Methods Approach to the Study of Pell Recipients' Formation of Academic Capital and the Influence of Institutional Agents Rowan University].
- Ruparelia, N. B. (2016). *Cloud computing*. Mit Press Cambridge, MA.
- Sadiku, M., Shadare, A. E., Musa, S. M., Akujuobi, C. M., & Perry, R. (2016). Data visualization. *International Journal of Engineering Research and Advanced Technology (IJERAT)*, 2(12), 11-16.
- Saini, C., & Arora, V. (2016). Information retrieval in web crawling: A survey. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI),
- Saini, R., & Behl, R. (2020). *An Introduction to AWS-EC2 (Elastic Compute Cloud)*. ICRMAT,
- Salem, F. (2016). *Comparative Analysis of Big Data Stream Processing Systems*
- Salinas, S. O., & Lemus, A. C. N. (2017). Data warehouse and big data integration. *AIRCC's International Journal of Computer Science and Information Technology*, 1-17.

- Salloum, S., Dautov, R., Chen, X., Peng, P. X., & Huang, J. Z. (2016). Big data analytics on Apache Spark. *International Journal of Data Science and Analytics*, 1, 145-164.
- Sangat, P., Indrawan-Santiago, M., & Taniar, D. (2018). Sensor data management in the cloud: Data storage, data ingestion, and data retrieval. *Concurrency and Computation: Practice and Experience*, 30(1), e4354.
- Saurkar, A. V., Pathare, K. G., & Gode, S. A. (2018). An overview on web scraping techniques and tools. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 4(4), 363-367.
- Savić, M., Ivanović, M., & Jain, L. C. (2019). *Complex networks in software, knowledge, and social systems*. Springer.
- Saxena, A., & Iyengar, S. (2020). Centrality measures in complex networks: A survey. *arXiv preprint arXiv:2011.07190*.
- Scherbakova, N. G., & Bredikhin, S. V. (2021). Co-authorship network structure analysis. *Journal of Physics: Conference Series*, 2099(1), 012055. <https://doi.org/10.1088/1742-6596/2099/1/012055>
- Shah, P. (2019). *Developing Big Data Analytics Architecture for Spatial Data*. PhD@VLDB,
- Shaikh, E., Mohiuddin, I., Alufaisan, Y., & Nahvi, I. (2019). Apache spark: A big data processing engine. 2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM),
- Shamrat, F., Tasnim, Z., Rahman, A. S., Nobel, N. I., & Hossain, S. A. (2020). An effective implementation of web crawling technology to retrieve data from the world wide web (www). *International Journal of Scientific & Technology Research*, 9(01), 1252-1256.
- Sharma, P. S., Yadav, D., & Thakur, R. (2022). Web page ranking using web mining techniques: a comprehensive survey. *Mobile Information Systems*, 2022, 1-19.
- Shi, Y., & Zhang, T. (2017). Design and implementation of a scalable distributed web crawler based on Hadoop. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA),
- Shin, H., Lee, K., & Kwon, H.-Y. (2022). A comparative experimental study of distributed storage engines for big spatial data processing using GeoSpark. *The Journal of Supercomputing*, 1-24.

- Silva, T. C., & Zhao, L. (2016). Complex Networks. 15-70. https://doi.org/10.1007/978-3-319-17290-3_2
- Singh, M. (2018). Virtualization in cloud computing-a study. 2018 international conference on advances in computing, communication control and networking (ICACCCN),
- Sirisuriya, D. S. (2015). A comparative study on web scraping.
- Sivaraj, V., Kangaialmal, A., & Kashyap, A. S. (2021). Enhancing fault tolerance using load allocation technique during virtualization in cloud computing. 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS),
- Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of business research*, 70, 263-286.
- Sporns, O. (2022). Graph theory methods: applications in brain networks. *Dialogues in Clinical Neuroscience*, 20(2), 111-121. <https://doi.org/10.31887/DCNS.2018.20.2/osporns>
- Srivastava, P., & Khan, R. (2018). A review paper on cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 8(6), 17-20.
- Sultan, N. A., & Abdullah, D. B. (2022). Scraping Google Scholar Data Using Cloud Computing Techniques. 2022 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM),
- Tavakkoli, M. (2019). Analyzing the Applicability of Kubernetes for the Deployment of an IoT Publish/Subscribe System.
- Thapa, A. B. (2022). Optimizing MongoDB performance with indexing: practices of indexing in MongoDB.
- Turban, E., Pollard, C., & Wood, G. (2018). *Information technology for management: On-demand strategies for performance, growth and sustainability*. John Wiley & Sons.
- Valavandan, R., Ghosh, S., Reddy, K., Parayatham, P., Sherif, U., Sharma, V., . . . Ambekar, N. Google Cloud Services for Collecting, Processing, Analyzing, and Visualizing the Types of COVID-19 Vaccines.
- Vanden Broucke, S., & Baesens, B. (2018). *Practical Web scraping for data science*. Springer.

- Vázquez Benítez, L. A. (2015). Computing methods for parallel processing and analysis on complex networks Universitat Politècnica de Catalunya].
- Veiga, J., Expósito, R. R., Pardo, X. C., Taboada, G. L., & Tourifio, J. (2016). Performance evaluation of big data frameworks for large-scale data analytics. 2016 IEEE International Conference on Big Data (Big Data),
- Welo, A. T., Kinzonzi, H. L., Khonde, N. B., & Mukendi, E. M. (2023). Big Data Architectures and Concepts. Journal of Innovation Information Technology and Application (JINITA), 5(2), 97-104.
- Yadav, C., Wang, S., & Kumar, M. (2013). Algorithm and approaches to handle large Data-A Survey. arXiv preprint arXiv:1307.5437.
- Yang, Y., Wang, X., Chen, Y., Hu, M., & Ruan, C. (2020). A novel centrality of influential nodes identification in complex networks. IEEE access, 8, 58742-58751.
- Zhang, L., Chen, H., & Li, Z. (2020). Research and Design of Inspection Cloud Platform Framework for Surveying and Mapping Products. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42, 57-61.
- Zomaya, A. Y. (2006). Handbook of nature-inspired and innovative computing: integrating classical models with emerging technologies. Springer Science & Business Media.

قرار المشرف

أشهد بأن إعداد الاطروحة الموسومة بـ " طريقة حوسبة البيانات الضخمة السحابية المتوازية لديناميكيات رأس المال الأكاديمي للبحث العلمي في العراق " قد جرى بإشرافي في جامعة الموصل/ كلية علوم الحاسوب والرياضيات وهي جزء من متطلبات نيل شهادة دكتوراه فلسفة في علوم الحاسوب.

التوقيع:

الاسم: أ.د. ضحى بشير عبدالله

التاريخ: / / ٢٠٢٤

إقرار الخبير اللغوي

أشهد بأن الاطروحة الموسومة بـ " نهج حاسوبي موازي للبيانات الضخمة السحابية لرأس المال الأكاديمي العراقي " قد تمت مراجعتها من الناحية اللغوية وتصحيح ما ورد فيها من أخطاء لغوية وتعبيرية وبذلك أصبحت مؤهلة للمناقشة بقدر تعلق الأمر لسلامة الأسلوب وصحة التعبير.

التوقيع:

الاسم: د. اوس عباس يونس

الكلية: الاداب

التاريخ: / / ٢٠٢٤

إقرار رئيس لجنة الدراسات العليا

بناءً على توصيتي المشرف وإقرار الخبير اللغوي، أرشح هذه الاطروحة للمناقشة.

التوقيع:

الاسم: أ.د. بيداء ابراهيم خليل

التاريخ: / / ٢٠٢٤

إقرار رئيس القسم

بناءً على توصيات المشرف وإقرار الخبير اللغوي وإقرار رئيس لجنة الدراسات العليا، أرشح هذه الاطروحة للمناقشة.

التوقيع:

الاسم: د. وائل وعدا الله محمود

التاريخ: / / ٢٠٢٤

قرار لجنة المناقشة

نشهد أننا أعضاء لجنة المناقشة قد اطلعنا على الاطروحة الموسومة " طريقة حوسبة البيانات الضخمة السحابية المتوازية لديناميكيات رأس المال الأكاديمي للبحث العلمي في العراق " وناقشنا الطالبة " نغم عجیل سلطان عبدالله " في محتوياتها وفيما له علاقة بها بتاريخ ٢٥ / ٤ / ٢٠٢٤ ونعتقد أنها جديرة بالقبول لنيل شهادة دكتوراه فلسفة في علوم الحاسوب.

التوقيع:
الاسم: د. نجلاء بديع ابراهيم
المرتبة العلمية: استاذ مساعد
التاريخ: ٢٠٢٤ / /
عضو اللجنة

التوقيع:
الاسم: د. محمد زكي حسن
المرتبة العلمية: استاذ مساعد
التاريخ: ٢٠٢٤ / /
عضو اللجنة

التوقيع:
الاسم: أ.د. ضحی بشیر عبدالله
المرتبة العلمية: استاذ
التاريخ: ٢٠٢٤ / /
عضو اللجنة (المشرف)

التوقيع:
الاسم: د. مرتضى محمد حمد
المرتبة العلمية: استاذ
التاريخ: ٢٠٢٤ / /
رئيس اللجنة

التوقيع:
الاسم: د. سجي جاسم محمد
المرتبة العلمية: استاذ مساعد
التاريخ: ٢٠٢٤ / /
عضو اللجنة

التوقيع:
الاسم: د. اشرف عبد المنعم عبد المجيد
المرتبة العلمية: استاذ مساعد
التاريخ: ٢٠٢٤ / /
عضو اللجنة

قرار مجلس الكلية

اجتمع مجلس كلية علوم الحاسوب والرياضيات بجلسته ()
المنعقدة بتاريخ ٢٥ / ٤ / ٢٠٢٤ وقرر منحه/ها شهادة دكتوراه فلسفة في علوم الحاسوب.

التوقيع:
الاسم: أ.د. ضحی بشیر عبد الله
التاريخ: ٢٠٢٤ / /
عميد كلية علوم الحاسوب والرياضيات

التوقيع:
الاسم: أ.د. صفوان عمر حسون
التاريخ: ٢٠٢٤ / /
مقرر مجلس كلية علوم الحاسوب والرياضيات

الخلاصة

إن الاستخدام المكثف للتقنيات الحديثة والإنترنت مع الأجهزة الذكية ينتج عنه بيانات واسعة النطاق تسمى البيانات الضخمة. إن تطور تقنيات الحوسبة السحابية يتيح التعامل مع البيانات الضخمة من حيث المعالجة والتخزين؛ لذلك، تم تطوير العديد من التقنيات من قبل الشركات والمؤسسات بهدف التوصل إلى أساليب فعالة. يمكن تنفيذ الأعمال السابقة التي تعمل على تحسين هذه الأساليب من خلال إشراك التقنيات المتقدمة مثل المعالجة المتوازية، والحاويات، والمحاكاة الافتراضية.

علاوة على ذلك، في علم الاجتماع، يمثل مفهوم رأس المال الاجتماعي السلوك الجماعي للفرد، والأعراف في المجتمع. وفي نفس السياق، يعني رأس المال الأكاديمي الأداء الجماعي للمؤلف أو المؤسسة. هذا المفهوم، لم تتم دراسته في الأدبيات السابقة. من ناحية أخرى، يعتبر Google Scholar محرك بحث الويب الأكثر شهرة للوصول إلى الأبحاث حول العالم ويمكن استخدام بياناته لتقييم الأداء الأكاديمي لمؤلف معين، أو جامعة، أو مؤسسة معينة من خلال الاستشهادات، ومؤشر h-index ، وما إلى ذلك. ومع ذلك، فإن معالجة هذه البيانات وتحليلها وتصورها تحتاج إلى اعتماد أساليب ديناميكية مخصصة. ولذلك، فإن تطوير نهج ديناميكي فعال لتحويل البيانات الضخمة إلى معرفة للتقييم من قبل المؤلفين أو الجامعات أمر بالغ الأهمية

تقدم هذه الأطروحة نظامًا ديناميكيًا من خلال اقتراح رانك جديد يسمى Academic Capital Distributed Network (ACDN). يقوم بجمع بيانات واسعة النطاق ديناميكيًا من Google Scholar باستخدام العديد من تقنيات الحوسبة السحابية. ومن ثم تمت معالجة البيانات التي تم جمعها وتصورها وتحليلها لتقييم الأداء الأكاديمي الديناميكي للمؤلفين والجامعات العراقية.

تم تصميم أدوات استخراج البيانات من الويب لجمع البيانات الديناميكية من Google Scholar للجامعات العراقية والمؤلفين بناءً على نطاقات جامعاتهم الرسمية (domain). تعتمد أدوات الكشط هذه على مفاهيم مستوحاة من تقنيات المعالجة المتوازية باستخدام أجهزة افتراضية مخصصة بشكل ديناميكي. يتم تخزين البيانات المجمعة بسلسلة باستخدام مجموعة النسخ المتماثلة في MongoDB Atlas لتسهيل إدارة البيانات بكفاءة وتوفير بنية تحتية ديناميكية قوية وقابلة للتطوير لتخزين وإدارة البيانات المجمعة. توفر هذه الطريقة توفرًا عاليًا وتسامحًا مع الأخطاء وقابلية التوسع للقراءة وتكامل البيانات. علاوة على ذلك، تم اعتماد بيئة حوسبة متوازية قائمة على السحابة لتسهيل التنظيم والمعالجة الفعالة للبيانات المجمعة، والاستفادة من أطر البيانات الضخمة، مع التركيز بشكل خاص على Apache Spark.

تم إنشاء مجموعتي بيانات جديدتين: واحدة للمؤلفين والأخرى للجامعات. باستخدام مجموعات البيانات هذه، تم إنشاء نموذجين للشبكة؛ الأول خاص بالجامعات العراقية؛ والمعروفة باسم شبكة الجامعات العراقية (IUN)، والأخيرة مخصصة للمؤلفين العراقيين؛ تُعرف باسم شبكة العلماء العراقيين (ISN). تم حساب رأس المال الأكاديمي للمؤلفين والجامعات باستخدام بيانات من نموذجي الشبكة IUN و ISN ومؤشرات Google Scholar. علاوة على ذلك، تم تقييم أداء المؤلفين والجامعات العراقية أيضًا، وتم تحديد الرتب بناءً على مقاييس رأس المال الأكاديمي الخاصة بهم والتي تتغير ديناميكيًا استجابةً للمؤشرات ومعايير البحث التي يستخدمها المؤلف أو الجامعة.

أظهرت النتائج الرئيسية أن Average Degree هو ١,٨، مما يعني أن كل باحث عراقي شارك في تأليف ورقتين كمتوسط. تعتبر هذه النتائج مقبولة مقارنة بشبكة ACM الدولية. من ناحية أخرى، فإن متوسط أقصر طول مسار لـ IUN هو ٣,٢ مما يعني أن التعاون العلمي بين الجامعات العراقية كان يشير إلى مسافة متوسطة بين العقد في المتوسط. ومع ذلك، فإن معامل التجمع لـ IUN هو ٠,٢٤٩، مما يعكس أن الجامعات العراقية ليست مرتبطة بقوة وأن الميل إلى التجمع معًا منخفض (بعيدًا عن ١). علاوة على ذلك، أظهرت نتائج رأس المال الأكاديمي للمؤلفين أن مؤلف من كلية علوم الحاسوب / جامعة بابل لديه أعلى رأس مال أكاديمي بين جميع المؤلفين في الجامعات العراقية. وكذلك رأس المال الأكاديمي الخاص بالجامعات العراقية أظهر أن جامعة بغداد حصلت على أعلى قيمة تليها الجامعة المستنصرية، جامعة البصرة، جامعة الموصل، جامعة بابل، وهكذا. وتتفق هذه النتيجة نسبيًا مع التصنيفات العالمية الأخرى مثل QS، Times، Webometrics. ويعتبر هذا العمل أول نوع من التحليل الذي يصنف جميع الجامعات العراقية والمؤلفين كمحور رئيسي، وهو ما يحظى باهتمام مسؤولي وزارة التعليم العالي والبحث العلمي في العراق.



وزارة التعليم العالي والبحث العلمي
جامعة الموصل
كلية علوم الحاسوب والرياضيات
قسم علوم الحاسوب

طريقة حوسبة البيانات الضخمة السحابية المتوازية لديناميكيات رأس المال الأكاديمي للبحث العلمي في العراق

اطروحة مقدمة
الى مجلس كلية علوم الحاسوب والرياضيات في جامعة الموصل
كجزء من متطلبات نيل شهادة دكتوراه فلسفة في
علوم الحاسوب

من قبل

نغم عجيل سلطان العجيلي

بإشراف
أ.د. ضحى بشير عبدالله البزاز