

Publication Automation System

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

N.S.S.Yaswanth (Register No. 38110355)

N.SunilVarma (Register No. 38110353)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **N.S.S.Yaswanth (38110355)**, **N.Sunilvarma (38110353)** who carried out the project entitled "**PUBLICATION AUTOMATION SYSTEM**" under my supervision from December 2021 to May 2022.

Internal Guide

Dr.G.Nagarajan

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,

Submitted for Viva-voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **N.S.S.YASWANTH** hereby declare that the project report entitled “**Publication Automation System**” done by me and **N.Sunil Varma** under the guidance of **Dr.G.Nagarajan** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.G. Nagarajan** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** were helpful in many ways for the completion of the project.

ABSTRACT

Despite the fact that their citation counts may be low, Researchers frequently examine Google Scholar (GS), Web of Science & Scopus, which are all used in research reviews.. From 252 Google Scholar subject categories there are 2,299 documents which have 2,448,055 citations in 2006. When GS, WoS and Scopus are compared then it is found that Among all areas, GS has the highest percentage of citations, approximately 67%-96% whereas Scopus has 35%-77% and WoS has 27%-73%. In GS, About 48% to 65% of citations came from non-journal sources such as books, conference papers, and unpublished papers. A further 19%-38% are non-English, and they tend to be less frequently cited than those in Scopus and WoS . So to extract the data from these publication sources we use web scrapping and web crawling. In this paper we will discuss The crawler shoots queries on the search interface by using keywords related to a user's topic of interest. This allows crawlers to gather links relevant to a particular domain without actually having to dig into the root domain of the domain. The proposed work provides the most pertinent information on a specific domain based on keywords without crawling through a large number of irrelevant links between them. The information from web is found by search engine. These search engines are of two types - human powered and crawler based. This user-generated search engine brings together a collection of high-quality websites that have been hand-picked by users. In a crawler-based search engine, three components are involved: A search-ranking algorithm, a crawler, and an indexer.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE-NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	INTRODUCTION	2
	1.1 DOMAIN DESCRIPTION	3
	1.1.1 WEB CRAWLER	3
	1.1.2 THE HISTORY OF WEB CRAWLER	3
	1.1.3 WORKING OF WEB CRAWLER	4
	1.1.4 CRAWLING TECHNIQUES	6
	1.1.5 PARALLEL CRAWLER	6
	1.1.6 ADVANTAGES OF WEB CRAWLING	7
	1.2 PROBLEM DEFINITION	8
	1.3 PROJECT DESCRIPTION	8
2	LITERATURE SURVEY	9
	2.1 AIM AND SCOPE	12
	2.2 EXISTING SYSTEM	12
3	METHODOLOGY	13
	3.1 PROPOSED SYSTEM	13
	3.2 IMPLEMENTATION	15
	3.3 SYSTEM DESIGN	16
	3.2.1 SYSTEM ARCHITECTURE	16
	3.2.2 UML DIAGRAMS	18
	3.2.3 USE CASE DIAGRAMS	19
	3.2.4 SEQUENCE DIAGRAM	20
	3.2.5 ACTIVITY DIAGRAM	22
	3.2.6 CLASS DIAGRAMS	22
	3.3 SYSTEM REQUIREMENTS	23

	3.3.1	HARDWARE SPECIFICATION	23
	3.3.2	SOFTWARE SPECIFICATION	24
	3.4	MODULES	25
	3.5	WORKING	25
	3.6	SOFTWARE DESCRIPTION	26
	3.7	LANGUAGES USED	31
	3.7.1	PYTHON	31
	3.7.1.1	PYTHON PROGRAMMING	
		CHARACTERISTICS	31
	3.7.1.2	APPLICATIONS OF PYTHON	
		PROGRAMMING	32
4		RESULTS AND DISCUSSION	34
5		CONCLUSION AND FUTURE WORK	36
	5.1	CONCLUSION	36
	5.2	FUTURE WORK	37
6		REFERENCES	37
		APPENDIX	
		A. SOURCE CODE	

LIST OF FIGURES

FIG NO.	NAME OF THE FIGURE	PAGE NO.
3.1	SYSTEM ARCHITECTURE	16
3.5	USE CASE DIAGRAM	20
3.6	SEQUENCE DIAGRAM	21
3.7	ACTIVITY DIAGRAM	22
7	OUT PUT	44

CHAPTER-1

INTRODUCTION

Google Scholar, which was launched in November 2004, transformed the way researchers and the general public searched for academic content by bringing the convenience of Google search to academic surroundings. Google Scholar was created as a search engine with the goal of determining the most pertinent publications for a particular query based on citation data . As of yet, the data of Google Scholar cannot be accessed in bulk. The free software Publish or Perish makes availability free citation of Google scholar data without database subscription. At present, Using third-party web scraping software is the only way to get more data from GS than Publish or Perish allows. Moreover The web repositories of scientific journals and agencies often provide data to funders and research institutions. Using Google Scholar data, researchers can analyze the performance of individual publications. Using these tools, stakeholders like search committees can find the most suitable candidate for a job. We provide a tool for verifying the publications of a group of researchers working in the same field. Online scrapers provide the needed data for the tool, The data for the tool is provided using a web-based user interface. We gathered the challenges we encountered scraping scientific web repositories throughout our research and summarized them here. Even when permission has been granted for data use, scraping is still necessary in some cases. During web crawling, A search engine is a piece of computer software that looks for information on the Internet. Using indexed databases, this program Looks for a result according to the user's query. Search engines use regularly updated indexes in order to cope with fast and efficient search results. These search engines keep their database index up to date by scouring a huge chunk of the internet. These search engines are different from web directories because directories are maintained by human editors whereas search engines use crawlers. Web spiders and web robots are terms used to describe web crawlers. These are computer viruses that recursively scan the internet using hyperlinks. Web crawling, often known as spidering, is the process of a crawler extracting data from the internet.

1.1 DOMAIN DESCRIPTION

1.1.1 WEB CRAWLER

A web crawler is a program/software or programmed script that browses the World Wide Web in a systematic, automated manner. The structure of the WWW is a graphical structure, i.e., the links presented in a web page may be used to open other web pages. Internet is a directed graph where webpage as a node and hyperlink as an edge, thus the search operation may be summarized as a process of traversing directed graph. By following the linked structure of the Web, web crawler may traverse several new web pages starting from a webpage. A web crawler move from page to page by the using of graphical structure of the web pages. Such programs are also known as robots, spiders, and worms. Web crawlers are designed to retrieve Web pages and insert them to local repository. Crawlers are basically used to create a replica of all the visited pages that are later processed by a search engine that will index the downloaded pages that help in quick searches. Search engines job is to storing information about several webs pages, which they retrieve from WWW. These pages are retrieved by a Web crawler that is an automated Web browser that follows each link it sees.

1.1.2 The History of Web Crawler

The first Internet “search engine”, a tool called “Archie” — shortened from “Archives”, was developed in 1990 and downloaded the directory listings from specified public anonymous FTP (File Transfer Protocol) sites into local files, around once a month [5], [6]. In 1991, “Gopher” was created, that indexed plain text documents. “Jughead” and “Veronica” programs are helpful to explore the said Gopher indexes [7], [8], [9], [10]. With the introduction of the World Wide Web in 1991 [11], [12] numerous of these Gopher sites changed to web sites that were properly linked by HTML links. In the year 1993, the “World WideWebWanderer” was formed the first crawler [13].

Although this crawler was initially used to measure the size of the Web, it was later used to retrieve URLs that were then stored in a database called Wandex, the first web search engine [14]. Another early search engine, “Aliweb” (Archie-Like Indexing for the Web) [15] allowed users to submit the URL of a manually constructed index of their site. The index contained a list of URLs and a list of user wrote keywords and descriptions. The network overhead of crawlers initially caused much controversy, but this issue was resolved in 1994 with the introduction of the Robots Exclusion Standard [16] which allowed web site administrators to block crawlers from retrieving part or all of their sites. Also, in the year 1994, “WebCrawler” was launched [17] the first “full text” crawler and search engine. The “WebCrawler” permitted the users to explore the web content of documents rather than the keywords and descriptors written by the web administrators, reducing the possibility of confusing results and allowing better search capabilities. Around this time, commercial search engines began to appear with [18], [19], [20], [21], [22], [23], [24] and [25] being launched from 1994 to 1997 [26]. Also introduced in 1994 was Yahoo! , a directory of web sites that was manually maintained, though later incorporating a search engine. During these early years Yahoo! and Altavista maintained the largest market share [26]. In 1998 Google was launched, quickly capturing the market [26]. Unlike many of the search engines at the time, Google had a simple uncluttered interface, unbiased search results that were reasonably relevant, and a lower number of spam results [27]. These last two qualities were due to Google’s use of the PageRank [28] algorithm and the use of anchor term weighting [29]. While early crawlers dealt with relatively small amounts of data, modern crawlers, such as the one used by Google, need to handle a substantially larger volume of data due to the dramatic enhance in the Web

1.1.3 Working of Web Crawler

The working of Web crawler is beginning with initial set of URLs known as seed URLs. They download web pages for the seed URLs and extract new links present in the downloaded pages. The retrieved web pages are stored and well indexed on the storage area so that by the help of these indexes they can later be retrieved as and when required. The extracted URLs from the downloaded page are confirmed to know whether their related documents have already been downloaded or not. If they are not downloaded, the URLs are again assigned to web crawlers for further downloading. This process is repeated till no more URLs are missing for downloading. Millions of pages are downloaded per day by a crawler to complete the target. Figure 2 illustrates the crawling processes.

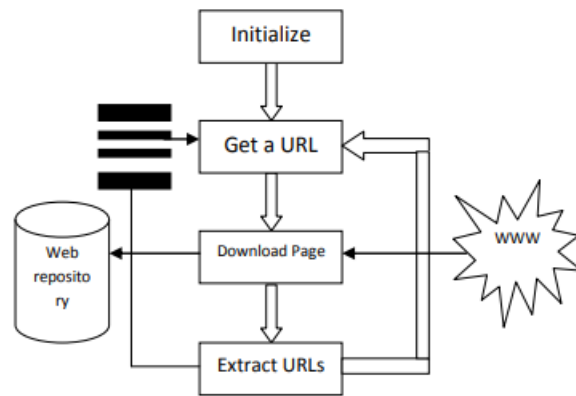


Figure 2: Flow of a crawling process

The working of a web crawler may be discussed as follows:

- Selecting a starting seed URL or URLs
- Adding it to the frontier → Now picking the URL from the frontier
- Fetching the web-page corresponding to that URL
- Parsing that web-page to find new URL links
- Adding all the newly found URLs into the frontier
- Go to step 2 and reiterate till the frontier is empty

Thus a web crawler will recursively keep on inserting newer URLs to the database repository of the search engine. So we can see that the major function of a web crawler is to insert new links into the frontier and to choose a fresh URL from the frontier for further processing after every recursive

step. To classify whether a transaction is fraud or legal, a large collection of transaction is required. The dataset is downloaded from Kaggle database. In this section the methodology followed is discussed in detail.

1.1.4 CRAWLING TECHNIQUES

There are a few crawling techniques used by Web Crawlers, mainly used are:

- A. General Purpose Crawling A general purpose Web Crawler collects as many pages as it can from a particular set of URL's and their links. In this, the crawler is able to fetch a large number of pages from different locations. General purpose crawling can slow down the speed and network bandwidth because it is fetching all the pages.
- B. Focused Crawling A focused crawler is designed to collect documents only on a specific topic which can reduce the amount of network traffic and downloads. The purpose of the focused crawler is to selectively look for pages that are appropriate to a pre-defined set of matters. It crawl only the relevant regions of the web and leads to significant savings in hardware and network resources.
- C. Distributed Crawling In distributed crawling, multiple processes are used to crawl and download pages from the Web.

1.1.5 PARALLEL CRAWLER

Now search engines do not depend on a single but on multiple web crawlers that run in parallel to complete the target. While functioning in parallel, crawlers still face many challenging difficulties such as overlapping, quality and network. Given below Figure illustrates the flow of multiple crawling processes.

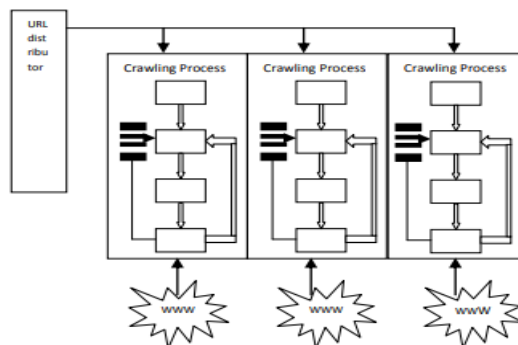


Figure 3: Flow of multiple crawling processes.

1.1.6 Advantages of Web Crawling

Keeping Tabs on Competitors This can be of great advantage especially for those dealing with tough competition in their industry. Sun Tzu, the Chinese general cum military strategist, opined that “If you know your enemies and yourself, you’ll never be defeated” To thrive in your industry, you need to survey your competitors. You need to get wind of what’s working for them. Their pricing, marketing strategies, and all.

With Web Crawlers, you can extract data automatically from various competitors’ websites without any hassle. This provides you and your employees the opportunity to save time for other productive tasks. The fact that the data are automatically extracted gives you the benefit of having access to data of great volume . If you have a sales team, product management team, or even the marketing team required to survey new products/services of competitors. Then you should consider Web Crawlers. It also gives the chance to review your pricing and ensure that they are competitive. With the extracted data you’ve acquired from various websites. You get to figure out the marketing strategies of your competitors.

1.2 PROBLEM DEFINITION

Faculties and students publishes research paper or journal based on the project and their point of interest from an organization. They can publish the paper in various websites such as Google scholar, web of science, scopus , leee, Springer etc. Publishing of paper from an organization increases the value for the organization and also for the author who published the paper. By using the web crawler technique we will classify the data based on the keyword given and also the paper which is published from a respective institute.

1.3 PROJECT DESCRIPTION

Web scraping is a technique for converting unstructured web data into structured data that can be stored and analyzed in a central database or spreadsheet. We use web scrapping for extracting the data from Google scholar and we will show the details obtained from the keyword given in a table format.

CHAPTER-2

LITERATURE SURVEY

Google Scholar, Web of Science, and Scopus: a systematic comparison of citations in 252 subject categories

Authors: Alberto Martín-Martín, Enrique Orduna-Malea , Mike Thelwall, Emilio Delgado López-Cózar, IEEE Access, Year: 2019

Despite citation counts from Google Scholar (GS), Web of Science (WoS), and Scopus being widely consulted by researchers and sometimes used in research evaluations, there is no recent or systematic evidence about the differences between them. In response, this paper investigates 2,448,055 citations to 2,299 English-language highly-cited documents from 252 GS subject categories published in 2006, comparing GS, the WoS Core Collection, and Scopus. GS consistently found the largest percentage of citations across all areas (93%-96%), far ahead of Scopus (35%-77%) and WoS (27%-73%). GS found nearly all the WoS (95%) and Scopus (92%) citations. Most citations found only by GS were from non-journal sources (48%-65%), including theses, books, conference papers, and unpublished materials. Many were non-English (19%-38%), and they tended to be much less cited than citing sources that were also in Scopus or WoS. Despite the many unique GS citing sources, Spearman correlations between citation counts in GS and WoS or Scopus are high (0.78-0.99). They are lower in the Humanities, and lower between GS and WoS than between GS and Scopus. The results suggest that in all areas GS citation data is essentially a superset of WoS and Scopus, with substantial extra coverage

Scraping Scientific Web Repositories: Challenges and Solutions for Automated

Content Extraction Authors: Philipp Meschenmoser ; Bela Gipp ; Norman

Meuschke ; 2015 10th International Conference on Computer Science &

Education (ICCSE) Aside from improving the visibility and accessibility of scientific publications, many scientific Web repositories also assess researchers' quantitative and qualitative publication performance, e.g., by displaying metrics such as the h-index. These metrics have become important for research institutions and other stakeholders to support impactful decision making processes such as hiring or funding decisions. However, scientific Web repositories typically offer only simple performance metrics and limited analysis options. Moreover, the data and algorithms to compute performance metrics are usually not published. Hence, it is not transparent or verifiable which publications the systems include in the computation and how the systems rank the results. Many researchers are interested in accessing the underlying scientometric raw data to increase the transparency of these systems. In this paper, we discuss the challenges and present strategies to programmatically access such data in scientific Web repositories. We demonstrate the strategies as part of an open source tool (MIT license) that allows research performance comparisons based on Google Scholar data. We would like to emphasize that the scraper included in the tool should only be used if consent was given by the operator of a repository. In our experience, consent is often given if the research goals are clearly explained and the project is of a non-commercial nature

Data Capturing: Methods, Issues and Concern

Authors: Afiqah Amirah Hamzah, Saiful Farik Mat Yatin, Nurul Athirah Ismail,

Siti Faridah Ghazali , Dr. C. S. Mala International Journal of Engineering

Research & Technology NCESC – 2018 Conference Proceedings Data capturing

is the method of putting a document into an electronic format. Many organizations

implement to automatically identify and classify information and make the available

within particular systems. It takes documents content, in any format, and converts it into

something that a computer can contrive. There have 2 methods to capture the data which are by manually or automated. Manual data entry from handwriting is very time consuming and prone to many errors when there is bulk amount of data is involved. Now day's organizations prefer using an automated data captured to convert text or handwriting from printed page into computer readable character. The method typically considered as part of data capturing include OCR, OMR, ICR, bar codes, QR codes and magnetic stripes. Automated data capturing is rapidly becoming an integral and necessary system in any organization. The system not only saves the time but also increase the speed and accuracy over manually entered data. This paper discussed the introduction of data capturing, methods, software, advantages and disadvantages and issues in data capture.

Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application.

Authors: Moaiad Khder

Web scraping or web crawling refers to the procedure of automatic extraction of data from websites using software. It is a process that is particularly important in fields such as Business Intelligence in the modern age. Web scraping is a technology that allow us to extract structured data from text such as HTML. Web scraping is extremely useful in situations where data isn't provided in machine readable format such as JSON or XML. The use of web scraping to gather data allows us to gather prices in near real time from retail store sites and provide further details, web scraping can also be used to gather intelligence of illicit businesses such as drug marketplaces in the darknet to provide law enforcement and researchers valuable data such as drug prices and varieties that would be unavailable with conventional methods. It has been found that using a web scraping program would yield data that is far more thorough, accurate, and consistent than manual entry. Based on the result it has been concluded that Web scraping is a highly useful tool in the information age, and an essential one in the modern fields. Multiple technologies are required to implement web scraping properly such as spidering and pattern matching which are discussed. This paper is looking into what web scraping is, how it works, web scraping stages, technologies, how it relates to Business Intelligence, artificial intelligence, data science, big data, cyber security, how it can be done with the Python language, some

of the main benefits of web scraping, and what the future of web scraping may look like, and a special degree of emphasis is placed on highlighting the ethical and legal issues.

2.1 AIM AND SCOPE

Previously if we want to get any information of the published journal we need to go the google scholar website and their we have to check about our required paper title, author

In this project using web scrapping we will classify the title of the page, author and url link of the paper.

2.2 EXISTING SYSTEM

There is no central server and this works in distributed environment. It has a distributed environment so it may not be aware about others collection, hence the problem of duplicity occurs. Due to this problem of duplicity more network bandwidth is consumed in this architecture.

CHAPTER-3

METHODOLOGY

3.1 PROPOSED SYSTEM

First we explore the crawling methodology before proposing our architecture to gain better understanding of the implementation details. As shown in Figure 1 Web Crawler has a typical life cycle which starts with set of the seed known as Uniform Resource Locator (URL), which is originally extracted from the web cache or the domain name server, and then digs deeper (Download) into the URL to fetch the sub URLs found into the main page. There are enormous issues and sophistications related to the web crawling development such as the network bandwidth and connection consumption, HTML tags and hyperlinks, web sites policies and regulations, storage and repositories, resources usage and scalability

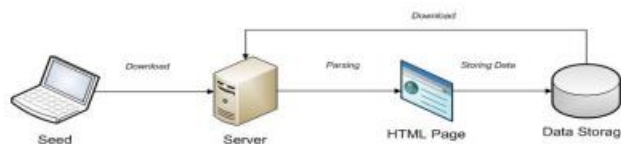


Fig. 1: Web Crawler Methodology

Figure 2 shows that any crawler contains what so-called as a (Frontier) which is basically a queue or a list that saves the unvisited URLs and usually followed by a frontier manager to define and organize the queue working mechanism. The user initializes the frontier with seed URLs to start with the crawling process. After the frontier manager passes the URLs for retrieving, it downloads the corresponding web pages and then extracts the entire sub URLs found in the page. The downloading process is accomplished using HyperText Transfer Protocol (HTTP) and recently the secured version of HTTP (HTTPS), whereby the crawler starts acting as a web client and sends HTTP requests to the server and reads the response contents. The crawler finishes the connection by setting a time out state, to ensure that the time is in control and not wasted in waiting for responses from crippled servers and put download restriction to prevent from reading large size web page.

Most of the frontiers in the web crawlers are placed in the main memory for faster data access and easy caching. The main issue with the frontiers is the URL priority assignment which affects how fast the frontier is filled up due to the huge number of the traversed web pages per unit of time. Thus, the crawler has to be equipped with efficient extraction method in order to fetch the URLs from the frontier manager in optimized manner.

After extracting the web pages and downloading them from the internet, the crawler starts the process of parsing and reading the contents to enable fetching the sub URLs and keeps the crawler in the extraction loop. Parsing can be as simple as extracting hyperlinks from the fetched web pages or it may involve more advanced techniques to analyze the HTML code such as linear scanning or deep tree traversing. Linear scanning is to split the HTML tags based on their functionality like, headers, forms, tables, paragraphs and so on. Deep tree traversing tends to break down the HTML code into a tree of objects to simplify the task of finding the next traversed URL.

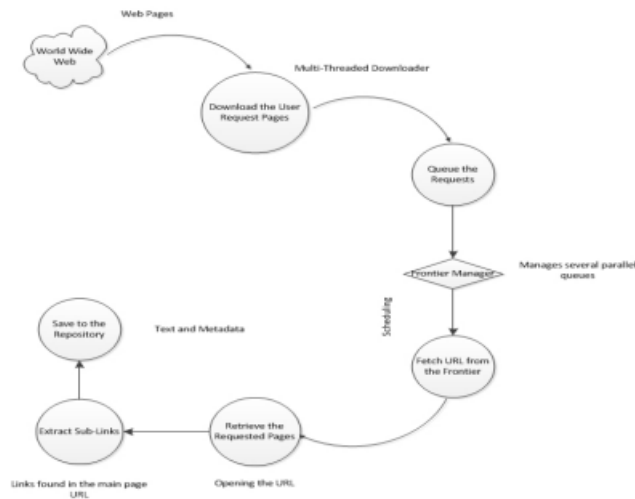


Fig.2: Data Flow for Basic Web Crawler

Process of converting URLs into a special form that is understood by internet routers, switches and Domain Name Servers (DNS) is one of web crawler processes and is called canonicalization. Conversion to the canonical form is independent process with somehow unique mechanism for each crawler. The user is free to specify what kind of rules and conditions the crawler will follow for canonicalization like protocol type and port number. However, crawlers may involve the heuristic approaches when such predefined and algorithmic approaches become impractical. In Figure 3 after canonicalization, resolving URLs into Internet Protocol (IP) addresses and pass it through the network to the DNS as a request to fetch web pages. It is found that, connection establishment between the web crawler, acting as a client, and DNS is the process that makes web crawler a fragile component.

The protocol type involved in the connection determines how fast the web pages extraction is. For example, using Unigram Data Protocol (UDP) avoids the troublesome of packet acknowledgment and what so-called three way handshake due to its connectionless property, while Transmission Connection Protocol (TCP) tend to acknowledge and commit three way handshake to open the transmission channel that result in connection overhead. Thus, it is recommended to use UDP to minimize this overhead, network congestion and memory conserve. In addition, asynchronous connection fosters the crawling when a single process is allowed to maintain hundreds of simultaneous connections.

3.2 Implementation

Figure 3 shows the overall flowchart for our web crawler architecture. The design of the web crawler depends exclusively on the used programming language as there are many libraries that simplify the implementation details. For example, Java, Perl, Python and C# provide application programming interfaces and libraries for opening URL stream, buffer reading, queue construction, input/output operations and pattern recognition. Thus, implementation tools play a significant role in designing and developing a customized web crawler.

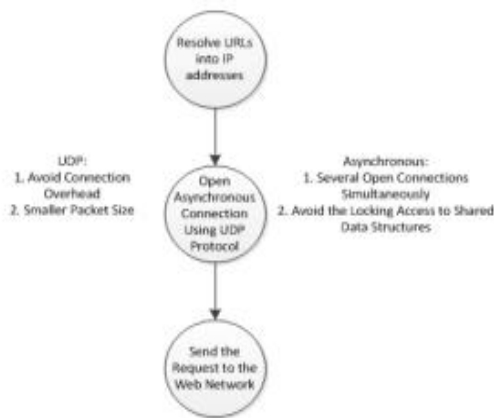


Fig.3: Canonicalization and IP Address Conversion

The first step in our implementation is to feed the crawler with URL input, and then we open an input stream based on that URL which is saved and added to the list by constructing a buffer to read the fed stream line by line. Append all the stream output to one string buffer to simplify the process of text manipulation and control over the attached strings using loops and control statements. All the URLs

are added to the linked list after testing certain conditions and constructing another linked list to save the visited URLs to avoid redundancy, hence it optimizes the retrieving process while fetching the URLs for page download. In the following, we explain the implementation details shown in Figure 4:

- The method `Fetch( )` is implemented with one parameters called `links` which provides the method with a list of URL's to start with
 - Crawler will start visiting the first URL and then extract the remaining list of URLs and queue them in `(ToVisit)` list
 - As long as the contents of URLs are eligible for politeness, protocol and type rules, the crawler will remain in the loop of extraction
 - To be more polite, we added the `(delay)` parameter to ensure that the crawler waits for the specified amount of seconds before revisiting the same URL, the more seconds the less the traffic you put on the server and vice versa
 - We provide additional control to the frontier manager to ensure that the crawler does not surf the entire web in an endless loop. `Depth( )` is a method to maintain the number of visited links within the same domain. `Breadth( )` is the other method that takes care of the other domains in the frontier manager
 - The `Breadth( )` method is capable of queuing the entire cache of web, but that will flood the memory and cause "Memory out of Buffer" error. That is where the priority check of the frontier comes in, whereby best first search is performed to pop the sub URLs in the frontier manager queue that has a limit of customized number. That was the overview of the details of implementing our focused crawler. What makes our crawler focused is the addition of dictionary that fills up the memory with words and phrases to compare against the extracted URLs

3.3 SYSTEM DESIGN

3.3.1 SYSTEM ARCHITECTURE:

The system proposed in this paper uses the algorithmic process for Web Crawling as in [21]. The most effective way to crawl a web is to access the pages in a depth first manner. This allows the crawled links to be acquired in a sequential hyperlink manner. Kumar et al. have proposed a parallel Depth First Search Algorithm in [21] which paves the way for the system in this paper. Depth First Search Algorithm allows the system to reach the hyperlinks associated with one page before moving on to the

next page. Also this system is developed on the backdrop of the Google idea in [22]. However the system takes a step forward by incorporating information integration along with Web Crawling so as to obtain processed data as required by most researchers. This process involves a concurrent working of Web Crawling and metatag extractor efficiently. The system uses the concept of metatag extraction to store the URL, title, keywords, and description in the database. This information can be obtained from the HTML content of every web page. In the future, it can be processed to store the body of the document after converting it to plain text if required by the application.

The features of the proposed system are as follows.

- (i) Identify and index the web through hyperlinks.
- (ii) Access new pages in the old sites through the hyperlinks.
- (iii) Store the metatag content in database for future use.
- (iv) Avoid calculating PageRank as it is time-consuming.
- (v) Alter robots to control the contents the system can access.

The entire Web Crawling system is depicted in the Block Diagram as shown in Figure [1](#). The block diagram consists of the process by which the documents are crawled from a web URL as specified by the analyser. The system proposed in this paper makes sure that it lies within the norms of Web Crawling Ethics.

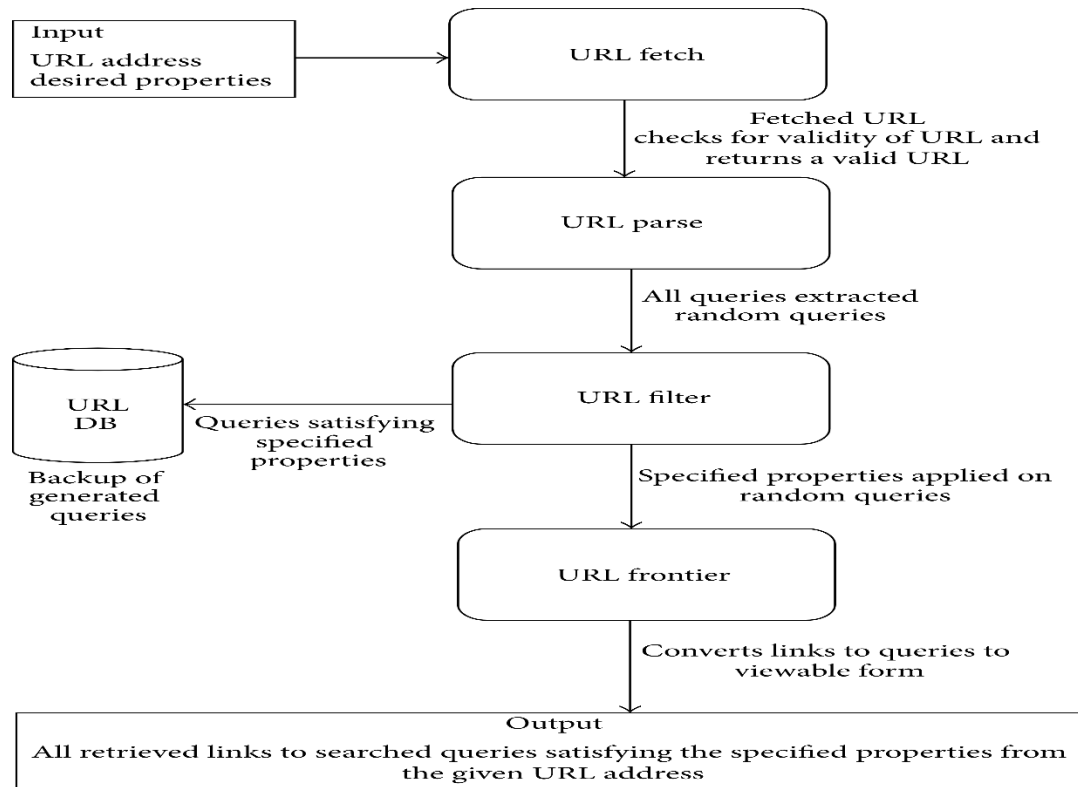


Fig.3.1 System Architecture

3.3.2 UML DIAGRAMS:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

3.3.3 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

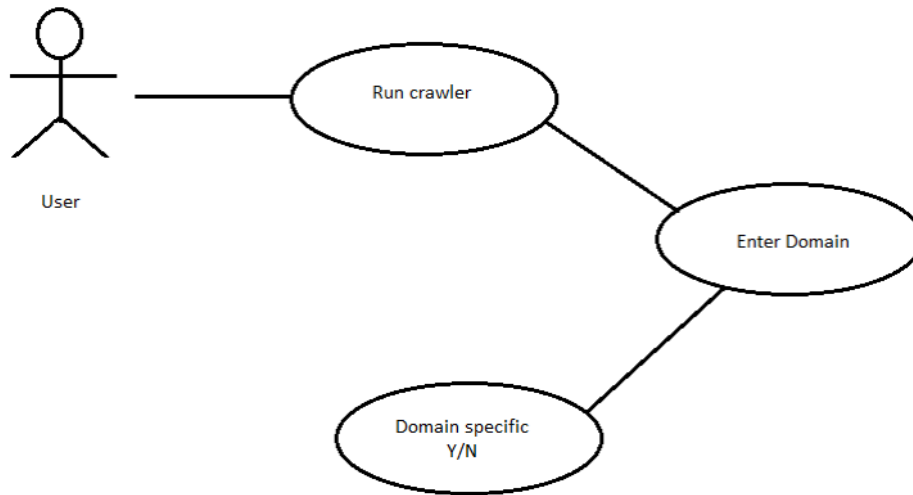


Fig.3.5 Use Case Diagram

3.3.4 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

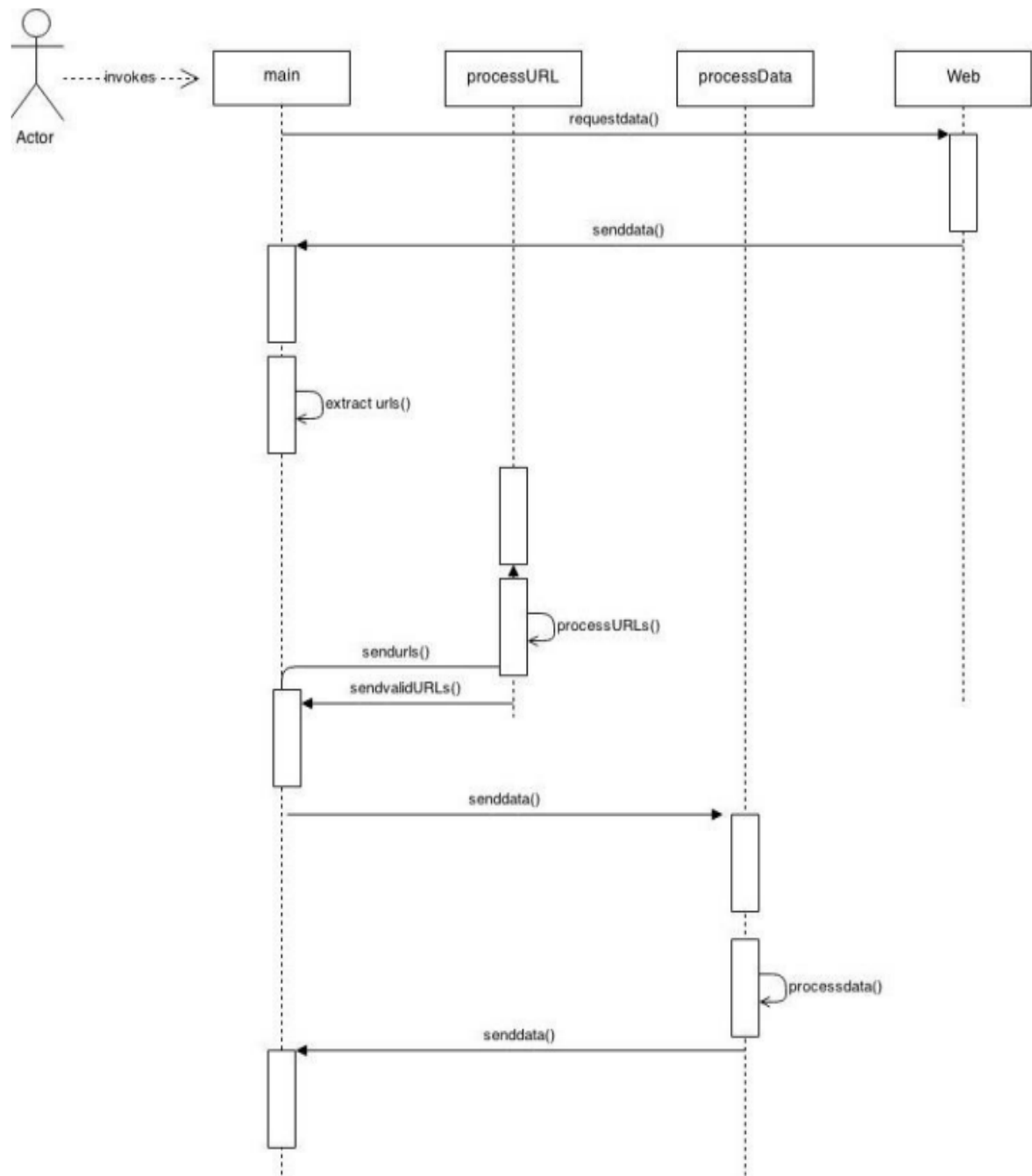


Fig.3.6 Sequence Diagram

3.3.5 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

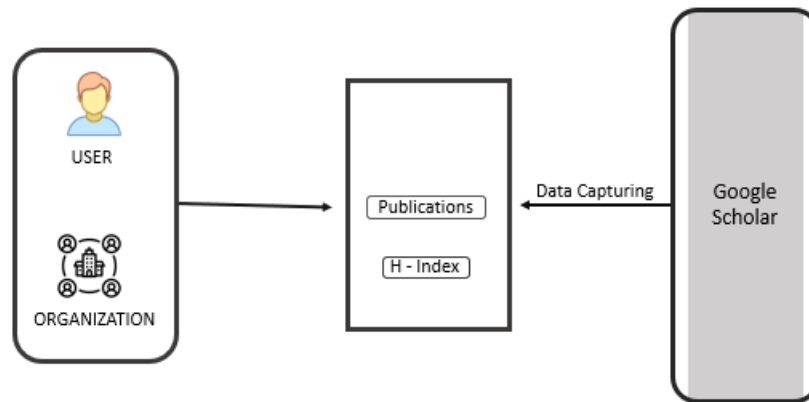


Fig.3.7 Activity Diagram

3.3.6 CLASS DIAGRAM:

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

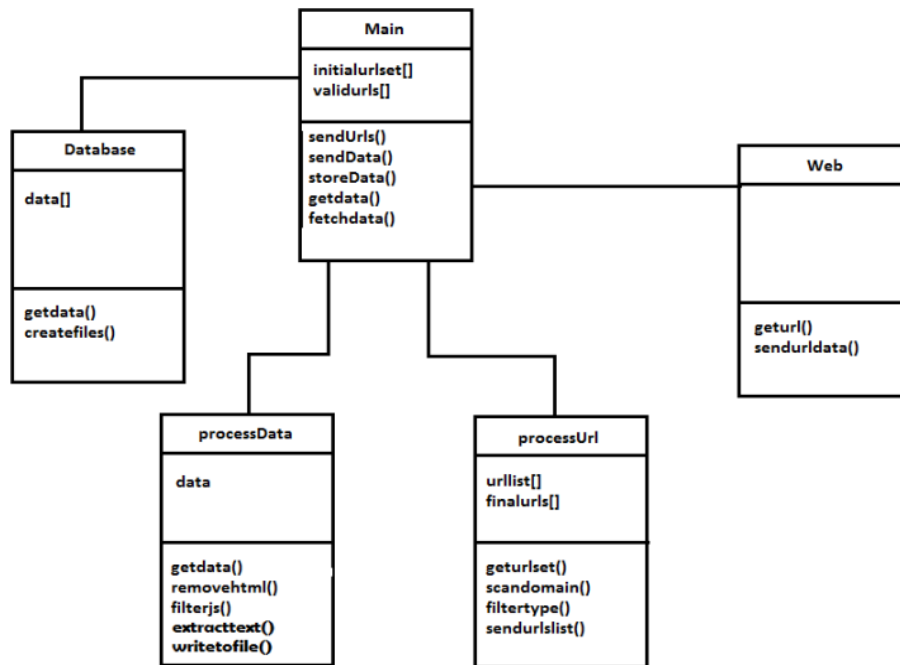


Fig.3.8 Class Diagram

3.4 SYSTEM REQUIREMENTS

3.4.1 Hardware Specification

- ✓ PROCESSOR : PENTIUM IV
- ✓ RAM : 8 GB
- ✓ PROCESSOR : 2.4 GHZ
- ✓ MAIN MEMORY : 8GB RAM
- ✓ PROCESSING SPEED : 600 MHZ
- ✓ HARD DISK DRIVE : 1TB
- ✓ KEYBOARD :104 KEYS

3.4.2 Software Specification

- ✓ FRONT END :PYTHON
- ✓ IDE : ANACONDA
- ✓ OPERATING SYSTEM :WINDOWS 10

3.5 MODULES

1. Data Preprocessing
2. Feature Extraction

Data Preprocessing

- a) It is a technique that is used to convert the raw data into a clean data set.
- b) In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Feature Extraction

In this module the data is extracted by the given key word and paper title, url and author of the paper is mentioned in the table.

3.6 WORKING:

The web scraping process is divided into 3 stages as shown in fig.2, which are:

Fetching stage: The desired website with the relevant information must first be accessed in what is known as the fetching phase this is accomplished via the HTTP protocol, which is an Internet protocol for sending and receiving requests. from web servers. Web browsers utilize similar methods to get material on web pages. In this step, libraries such as curl 2 and wget 3 can be used by sending an HTTP GET request to the target address (URL) and get the HTML page as a response (Persson, 2019)

Extraction stage: After retrieving the HTML page, the important data should be extracted. Regular expressions, HTML parsing libraries, and XPath queries are utilized in this step, which is referred to as the extraction stage. The XML Path Language (XPath) is a tool for finding information in documents. This is the second phase of the project. (Persson, 2019)

Transformation stage: Now that just the relevant data remains, it may be converted into a structured

format for presentation or storage. Using the stored data, we can gather information which can help the business intelligence in making a better decision and much more. (Persson, 2019)

3.7 SOFTWARE DESCRIPTION

- *Installing Anaconda on Windows*
- **DOWNLOADING AND INSTALLATION STEPS FOR Installing Anaconda on Windows**
- **Follow the steps below to install the Anaconda distribution of Python on Windows.**

Steps:

1. [Download the Anaconda installer.](#)



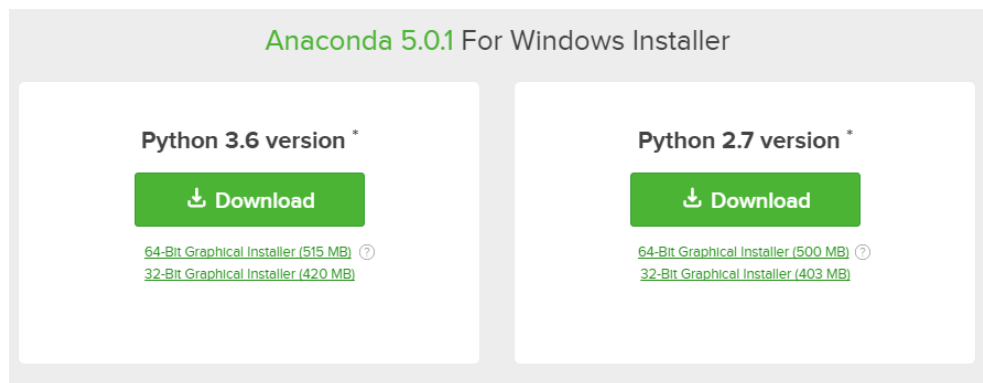
Select Windows

Select Windows where the three operating systems are listed.



Download

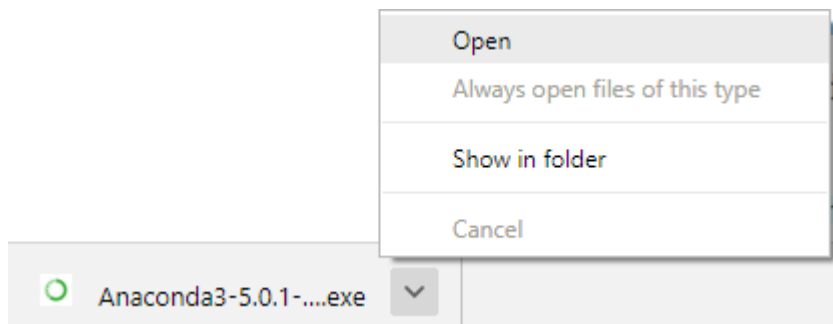
Download the most recent Python 3 release. At the time of writing, the most recent release was the Python 3.6 Version. Python 2.7 is legacy Python. For problem solvers, select the Python 3.6 version. If you are unsure if your computer is running a 64-bit or 32-bit version of Windows, select 64-bit as 64-bit Windows is most common.



2. Double click the installer to launch.

Open and run the installer

Once the download completes, open and run the .exe installer



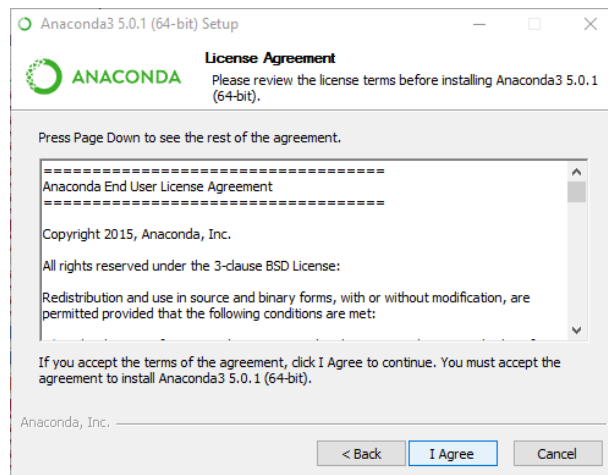
3. Click Next.

At the beginning of the install, you need to click **Next** to confirm the installation.



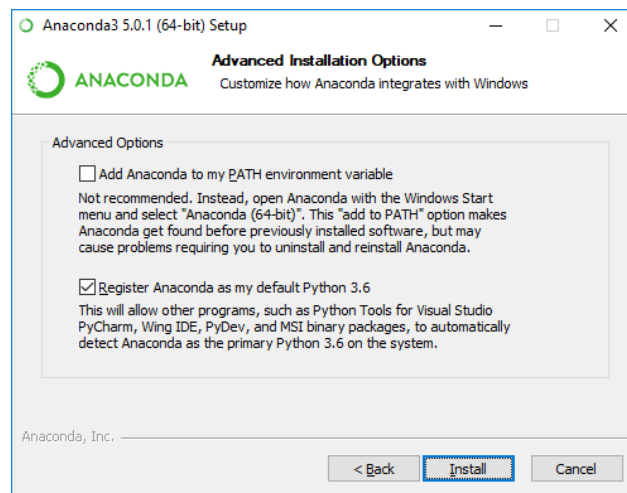
4. Read the licensing terms and click “I Agree”.

Then agree to the license.

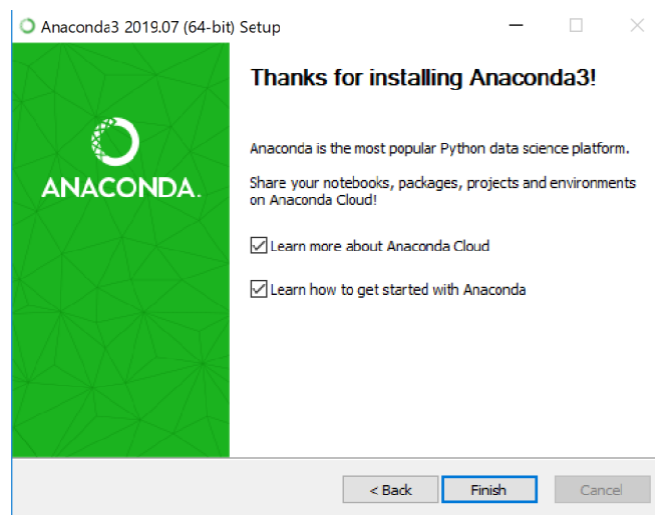


5. Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.

At the Advanced Installation Options screen, I recommend that you **do not check** “Add Anaconda to my PATH environment variable”



6. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:



If you wish to read more about Anaconda Cloud and how to get started with Anaconda, check the boxes “Learn more about Anaconda Cloud” and “Learn how to get started with Anaconda”. Click the Finish button.

- [Orange 3 App](#)
- [Rstudio](#)

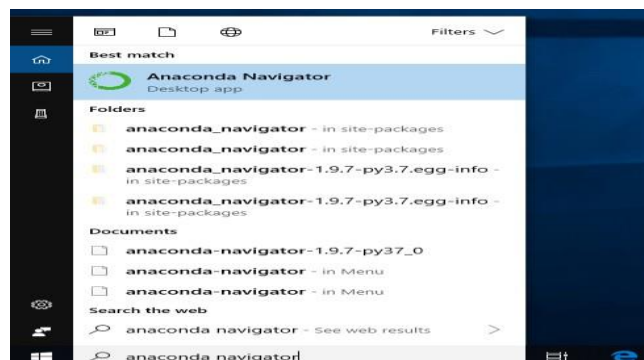
Verifying your installation: You can confirm that Anaconda is installed and working with Anaconda Navigator or conda.

Anaconda Navigator: Anaconda Navigator is a graphical user interface that is automatically installed with Anaconda. Navigator will open if the installation was successful. If Navigator does not open, review our help resources.

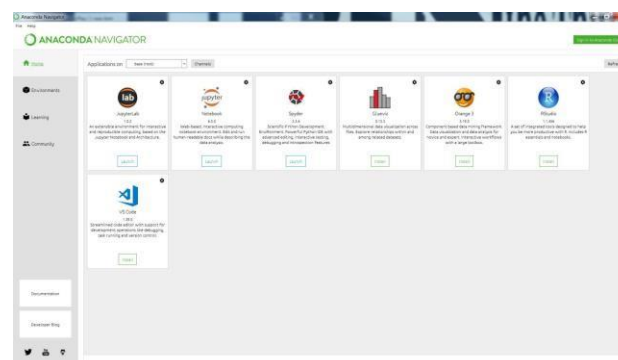
Windows: Click Start, search or select Anaconda Navigator from the menu.

MacOS: Click Launchpad, select Anaconda Navigator. Or, use Cmd+Space to open Spotlight Search and type “Navigator” to open the program.

Linux: See next section.



Getting started with Navigator



The following applications are available by default in Navigator:

- [JupyterLab](#)
- [Jupyter Notebook](#)
- [Spyder](#)
- [VSCode](#)
- [Glueviz](#)

3.8 LANGUAGES USED

3.8.1 *Python Language*

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

3.8.1.1 Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions

- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

3.8.1.2 Applications of Python Programming

3.8.1.2.1 Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

3.8.1.2.2 Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

3.8.1.2.3 Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

3.8.1.2.4 Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five

typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Beautiful Soap:

Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

Beautiful Soup was started by Leonard Richardson, who continues to contribute to the project, and is additionally supported by Tidelift, a paid subscription to open-source maintenance

It is available for Python 2.7 and Python 3

CHAPTER-4

RESULTS AND DISCUSSION

In web scraping, code reuse and maintenance are especially critical. Code reuse is important in web scraping because when we develop program to mostly have some common feature that we used before. For example, for website data scraping, the program needs to access the website so if you built a good code to access a site using username and password, you can reuse it in a future project if you are going to run it in the same situation. Web scraping involves code maintenance because, no matter how reliably you construct your scraper, a significant change in the targeted site's design or behavior might make it extremely difficult for your program to find the appropriate data on the site. If you have a web scraping.

It was discovered that the web scraping has a wide variety of applications across the field of Business Intelligence. Using web scraping programs would involve a higher upfront cost, but would pay for themselves over and over again as a form of short-term loss, long term gain. Furthermore, using a web scraping program would yield data that is far more thorough, accurate, and consistent than manual entry. However, it is important to use it in tandem with other data

processing tools, as the raw data on its own will have far too many extraneous details to be useful unless properly filtered and processed.

Although online scraping is an effective method for obtaining massive data sets, it has some ethical and legal consideration, scrapping may happen over copyrighted data and lead to copyright infringement furthermore aggressive spidering usually leads to large number of requests to sites that may cause loss of services which is naturally undesirable by site owners procedure that runs on a regular basis for any length of time, you'll need to be ready to make code changes to compensate for changes on the destination web site. Finally, Python is perfect for easy implementation. Web scraping is frequently required in order to go to more interesting aspects of a project. Simple web scraping in Python can be pulled up fast, especially if you have a large library of reusable code. Based on the literature reviewed web scrapping is a valuable tool that can allow us to extract meaningful knowledge from otherwise unstructured data. Combining web scrapping with other technologies such as big data can yield further value when performing data extraction in scale

However, there has been controversies surrounding web scraping as users have plagiarized whole websites and claimed the data as their own, resulting in lawsuits and copyright infringements. To avoid any further conflicts in the future, any company or organization that engage in web scraping must be under surveillance or be subjected to some kind of process that can detect plagiarism. Natural Language Processing (NLP) is another way web scrapping can be improved with libraries included in many computer languages that can be used for data cleaning, filtering, organizing such as Python and R languages. Using web scraping is highly recommended as it is cheaper as compared to other methods of data collection given that it is done in an ethical manner and no one is harmed.

CHAPTER-5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Web scraping is a highly useful tool in the information age, and an essential one in different fields for any company wishing to maintain their online presence, which in of itself is very necessary to survive in today's market. While more strict legal measures may be enforced in the coming years, the rate of this new market will keep continuing to grow, and it is therefore an incredibly valuable skill to learn. One of the best languages for implementing it in would be Python, due to its fast learning curve and powerful syntax. We distinguished three categories of challenges to scraping: size limitations of the result sets, dynamic content, and access barriers. Size limitations of the result sets can often be addressed by carefully analyzing the website's structure and techniques employed to reduce server load. Accessing dynamic content commonly requires the use of headless Web browsers running full JavaScript engines, which imposes a large performance overhead onto the system and drastically reduces the crawling speed. Avoiding access barriers is of variable difficulty. Switching IP addresses is an easy solution but usually involves using for. This solution puts a significant load on the or network and causes many harmless nodes to be flagged as bots, which impacts private browsing for human users. Google Scholar is a prodigious mine of academic information that covers all fields of knowledge.

Thanks to GS we now have access to previously unexplored territories of knowledge which, even if only roughly, are allowing us to form a broader mental picture of academic activity. This platform sheds light where previously there was only darkness. Google Scholar can help to definitely open the academic Pandora's box [4]. Opening this box will bring to light document typologies (especially monographs, theses, reports, conference communications, and book chapters) from the scientific core and periphery, which were previously invisible and inaccessible. The bibliographic features of these documents will probably change certain axioms and prejudices of academic evaluation.

5.2 FUTURE WORK

In future this project can be further modified and can be used to classify the data according to the given name not only from Google scholar but also we can implement it to extract the data from various publication sites. This can be useful in NAAC ranking for a college. We can divide the number the papers published by the faculty from respective institute and we can display in our website

6 REFERENCES

- [1] Brin, S., Page, L. (2012) "Reprint of: The anatomy of a large-scale hypertextual web search engine." *Comput. Networks*. 56 (18): 3825–3833. doi:10.1016/j.comnet.2012.10.007.
- [2] Kumar, M., Bhatia, R., Rattan, D. (2017) "A survey of Web crawlers for information retrieval." *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* e1218. doi:10.1002/widm.1218.
- [3] Shokouhi M, Chubak P, Raeesy Z. (2005) "Enhancing focused crawling with genetic algorithms." In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on 2005, IEEE Apr 4, 2: 503-508.*
- [4] Chakrabarti, S., Van Den Berg, M., Dom, B. (1999) "Focused crawling: A new approach to topic-specific Web resource discovery." *Comput. Networks*. 31 (11), 1623–1640. doi:10.1016/S1389-1286(99)00052-3.
- [5] Ipeirotis, P.G., Agichtein, E., Jain, P., Gravano, L. (2007) "Towards a query optimizer for text-centric tasks". *ACM Trans. Database Syst.* 32 (4): 21 doi:10.1145/1292609.1292611.
- [6] Priyatam PN, Vaddepally SR, Varma V. (2012) "Domain specific search in indian languages." In *Proceedings of the first workshop on Information and knowledge management for developing*

region 2012 Nov 2, ACM: 23-30.

[7] Tang, T.T., Hawking, D., Craswell, N. and Griffiths, K. (2005) "Focused crawling for both topical relevance and quality of medical information." In Proceedings of the 14th ACM international conference on Information and knowledge management, October 2005, ACM: 147-154.

[8] Altıngözü İS, Ulusoy O. (2004) "Exploiting interclass rules for focused crawling." IEEE Intelligent Systems. 2004 Nov;19 (6):66-73.

[9] Kumar M, Bhatia R, Ohri A, Kohli A. (2016) "Design of focused crawler for information retrieval of Indian origin Academicians." In Advances in Computing, Communication, & Automation (ICACCA)(Spring), International Conference on 2016 Apr 8, IEEE:1-6.

[10] Zhao, F., Zhou, J., Nie, C., Huang, H., & Jin, H. (2016). SmartCrawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces. IEEE transactions on services computing, 9(4), 608-620.

[11] A. Balla, A. Stassopoulou, and M. D. Dikaiakos. Real-time Web Crawler Detection. In Proc. 18th Int. Conf. on Telecommunications (ICT), pages 428-432, 2011.
<http://doi.org/10.1109/CTS.2011.5898963>

[12] J. Beel and B. Gipp. Google Scholar's Ranking Algorithm: The Impact of Articles' Age (An Empirical Study). In Proc. 6th Int. Conf. on Information Technology: New Generations (ITNG), pages 160-164, 2009. <http://doi.org/10.1109/ITNG.2009.317>

[13] C. Bomhardt, W. Gaul, and L. Schmidt Thieme. Web Robot Detection — Preprocessing Web

Logfiles for Robot Detection. In New Developments in Classification and Data Analysis: Proc. of the Meeting of the Classification and Data Analysis Group of the Italian Statistical Society, pages 113-124. Springer, 2005. http://doi.org/10.1007/3-540-27373-5_14

[14] M. Dikaiakos, A. Stassopoulou, and L. Papageorgiou. Characterizing Crawler Behavior from Web Server Access Logs, pages 369-378. Springer, 2003.
http://doi.org/10.1007/978-3-540-45229-4_36

[15] D. Doran and S. S. Gokhale. Web Robot Detection Techniques: Overview and Limitations. Data Mining and Knowledge Discovery, 22 (1): 183-210, 2011.
<http://doi.org/10.1007/s10618-010-0180-z>

APPENDIX

A. SOURCE CODE

```
import requests

headers = {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36'}

url = 'https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=G.nagarajan+sathyabama+university&oq='
response=requests.get(url,headers=headers) response.status_code
200

page_contents = response.text len(page_contents)
152913
```

Use BeautifulSoup to parse and extract information

- Parse and explore the structure of downloaded web pages using Beautiful soup.
- Use the right properties and methods to extract the required information.
- Create functions to extract from the page into lists and dictionaries.
- (Optional) Use a REST API to acquire additional information if required.

```
!pip install beautifulsoup4
```

Requirement already satisfied: beautifulsoup4 in c:\users\lenovo\ anaconda3\lib\site-packages (4.9.3)

Requirement already satisfied: soupsieve>1.2 in c:\users\lenovo\ anaconda3\lib\site-packages
(from beautifulsoup4) (2.2.1) from bs4 import BeautifulSoup

```
doc = BeautifulSoup(page_contents,'html.parser') type(doc)
```

```
bs4.BeautifulSoup
```

```
paper_names_tag = doc.select('[data-lid]')
```

```
paper_names = [] for tag in paper_names_tag:
```

```
paper_names.append(tag.select('h3')[0].get_text()) paper_names
```

['Hybrid intrusion detection system for private cloud: a systematic approach',
 'Brain computer interface for smart hardware device',
 'Hybrid genetic algorithm for medical image feature extraction and selection',
 'Design of interleaved flyback converter',
 '[PDF][PDF] Performance analysis of voltage profile, power, angle of injection using combined FACTS device',
 '[PDF][PDF] Power quality improvement for thirty bus system using UPFC and TCSC',
 'Wireless soil monitoring sensor for sprinkler irrigation automation system',
 'A fuzzy based Co-Operative Blackmailing Attack detection scheme for Edge Computing nodes in MANET-IOT environment',
 '[PDF][PDF] Rule-based semantic content extraction in image using fuzzy ontology',
 '[PDF][PDF] Semantically effective visual concept illustration for images']

```

citations = doc.select("[title=Cite] + a") links = paper_names_tag[0].find('h3')
link_tag = doc.find_all('h3',{ "class" : "gs_rt"}) link_tag[0]
<h3 class="gs_rt" ontouchstart="gs_evt_dsp(event)"><a data-
clk="hl=en&sa=T&ct=res&cd=0&d=14641454419450029739&
;ei=U7BSYtLuEJaM6rQPISayA8" data-clk-atid="q2ZkgjIIMsJ"
href="https://www.sciencedirect.com/science/article/pii/S1877050915006985"
id="q2ZkgjIIMsJ">Hybrid intrusion detection system for private cloud: a systematic
approach</a></h3> link_tag[0].a["href"]

```

```

'https://www.sciencedirect.com/science/article/pii/S1877050915006985' links = []

```

```

for i in range(len(link_tag)) : links.append(link_tag[i].a["href"])

```

```

links

```

```

['https://www.sciencedirect.com/science/article/pii/
S1877050915006985',

```

```

'https://content.iospress.com/articles/international-journal-of-rftechnologies/rft180167',

```

```

'https://www.sciencedirect.com/science/article/pii/S1877050916305324',

```

```

'https://www.inderscienceonline.com/doi/abs/10.1504/IJIE.2019.100044',

```

'https://www.researchgate.net/profile/Murugan-Arumugham/publication/289063753_Performance_analysis_of_voltage_profile_power_angle_of_injection_using_combined_facts_device/links/5a20e5b6a6fdcccd30e042cf/Performance-analysis-of-voltage-profile-power-angle-of-injection-using-combined-facts-device.pdf',
 'https://www.academia.edu/download/34901386/46766-sep14.pdf',
 'https://link.springer.com/article/10.1007/s11277-017-4948-y',
 'https://www.sciencedirect.com/science/article/pii/S0167739X21002478',
 'https://www.researchgate.net/profile/Thyagarajan-K-K/publication/280313360_Rule-Based_Semantic_Content_Extraction_in_Image_Using_Fuzzy_Ontology/links/55b1eda008ae092e96500569/Rule-Based-Semantic-Content-Extractionin-Image-Using-Fuzzy-Ontology.pdf', 'http://www.ijfcc.org/papers/282-M019.pdf'] **authors of the paper**
 authors_tag = doc.find_all("div", {"class": "gs_a"}) len(authors_tag)
 10 authors_tag[0]
 <div class="gs_a">PK Rajendran, B Muthukumar, G Nagarajan -
 Procedia Computer Science, 2015 - Elsevier</div> authors_tag[4].text
 'A Murugan, G Nagarajan, SDS Jebaseelan... \xa0- Int. J. Appl\xa0..., 2014 - researchgate.net' x = authors_tag[0].find('a').text x
 'PK Rajendran'
 authors_list = [] **for** i **in** authors_tag:
 authors_list.append(i.text) authors_list
 ['PK Rajendran, B Muthukumar, G Nagarajan\xa0- Procedia Computer Science, 2015 - Elsevier',
 'G Nagarajan, RI Minu... \xa0- International Journal of\xa0..., 2019 - content.iospress.com',
 'G Nagarajan, RI Minu, B Muthukumar... \xa0- Procedia Computer\xa0..., 2016 - Elsevier',
 '...', DG Immanuel, G Nagarajan\xa0- ... \xa0Journal of Intelligent\xa0..., 2019 - inderscienceonline.com',
 'A Murugan, G Nagarajan, SDS Jebaseelan... \xa0- Int. J. Appl\xa0..., 2014

- researchgate.net',
 '...', B Padmanabhan, G Nagarajan\xa0- Indian Journal of\xa0..., 2014 - academia.edu',
 'G Nagarajan, RI Minu\xa0- Wireless Personal Communications, 2018 - Springer',
 'SV Simpson, G Nagarajan\xa0- Future Generation Computer Systems,
 2021 - Elsevier',
 'G Nagarajan, KK Thyagarajan\xa0- Int Rev Comput Softw, 2014 - researchgate.net',
 'KK Thyagarajan, G Nagarajan\xa0- International Journal of Future Computer\xa0..., 2014 - ijfcc.org']

Create CSV file(s) with the extracted information import pandas as pd

```
paper_dict = {
    'paper title ' : paper_names,
    'authors' : authors_list,
    'url of paper' : links
}

papers_df = pd.DataFrame(paper_dict) papers_df
```

OUT

In [33]: papers_df

Out[33]:

	paper title	authors	url of paper
0	Hybrid intrusion detection system for private ...	PK Rajendran, B Muthukumar, G Nagarajan - Proc...	https://www.sciencedirect.com/science/article/...
1	Brain computer interface for smart hardware de...	G Nagarajan, RI Minu... - International Journal ...	https://content.iospress.com/articles/internat...
2	Hybrid genetic algorithm for medical image fea...	G Nagarajan, RI Minu, B Muthukumar... - Procedia...	https://www.sciencedirect.com/science/article/...
3	Design of interleaved flyback converter	..., DG Immanuel, G Nagarajan - ... Journal of Int...	https://www.inderscienceonline.com/doi/abs/10....
4	[PDF][PDF] Performance analysis of voltage pro...	A Murugan, G Nagarajan, SDS Jebaseelan... - Int....	https://www.researchgate.net/profile/Murugan-A...
5	[PDF][PDF] Power quality improvement for thirt...	..., B Padmanabhan, G Nagarajan - Indian Journal...	https://www.academia.edu/download/34901386/467...
6	Wireless soil monitoring sensor for sprinkler ...	G Nagarajan, RI Minu - Wireless Personal Commu...	https://link.springer.com/article/10.1007/s112...
7	A fuzzy based Co-Operative Blackmailing Attack...	SV Simpson, G Nagarajan - Future Generation Co...	https://www.sciencedirect.com/science/article/...
8	[PDF][PDF] Rule-based semantic content extract...	G Nagarajan, KK Thyagarajan - Int Rev Comput ...	https://www.researchgate.net/profile/Thyaghara...
9	[PDF][PDF] Semantically effective visual conce...	KK Thyagarajan, G Nagarajan - International J...	http://www.ijfcc.org/papers/282-M019.pdf



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in



National Conference on Computational Intelligence and Communication Networks



NCCICN

24TH – 25TH MARCH 2022



Certificate of Presentation

*This is to certify that Dr./Mr./Ms. **NARAYANAM YASWANTH**, of **Sathyabama Institute of Science and Technology**, has presented a paper entitled **Publication Automation System**, in the National Conference on Computational Intelligence and Communication Networks (NCCICN 2022).*

Dr. T. Sasikala

Conference Chair, Professor & Dean
School of Computing

Dr. L. Lakshmanan

Convener
Professor & Head, CSE

Dr. S. Vigneshwari

Convener
Professor & Head, CSE



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
[DEEMED TO BE UNIVERSITY]

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in



National Conference on Computational Intelligence and Communication Networks



NCCICN

24TH – 25TH MARCH 2022



Certificate of Presentation

*This is to certify that Dr./Mr./Ms. **NANDYALA SUNIL VARMA**, of **Sathyabama Institute of Science and Technology**, has presented a paper entitled **Publication Automation System**, in the National Conference on Computational Intelligence and Communication Networks (NCCICN 2022).*

Dr. T. Sasikala

Conference Chair, Professor & Dean
School of Computing

Dr. L. Lakshmanan

Convener
Professor & Head, CSE

Dr. S. Vigneshwari

Convener
Professor & Head, CSE