International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 5, October 2024, pp. 5468~5480
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i5.pp5468-5480

Developing an effective focused crawler to retrieve data of Indian-origin scientists and utilizing text classification for comparative analysis

Shivani Gautam[1], Rajesh Bhatia[2], Shaily Jain[3]
[1]Chitkara University School of Engineering and Technology, Chitkara University, Himachal Pradesh, India
[2]Department of Computer Science and Engineering, Punjab Engineering College (PEC) University of Technology, Chandigarh, India
[3]Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

Article Info

Article history:

Received Mar 20, 2024
Revised Jun 24, 2024
Accepted Jul 2, 2024

Keywords:

Comparative analysis
Focused web crawler
Indian scientists database
natural language processing
text classification
Information retrieval
Web scraping

ABSTRACT

This article presents the implementation of focused web crawling to retrieve data about scientists of Indian ancestry who are working in foreign nations. This study demonstrates the effectiveness of web scraping in obtaining large amounts of data from publicly available online pages. The objective is to construct a collection of data pertaining to Indian scientists who are now employed in national laboratories overseas. Collecting a vast quantity of data on the aforementioned Indian scientists through manual search is a pointless task. Therefore, this study proposes a detailed plan for a focused web crawler that can gather similar data. Subsequently, we present a comprehensive assessment of numerous classification models on this newly created dataset. Our assessments indicate that the random forest model surpasses the other supervised models. The empirical findings on large datasets demonstrated that the combination of random forest with synthetic minority oversampling technique (SMOTE) and k-fold cross-validation methods yielded better performance compared to K-nearest neighbors (KNN), support vector machine (SVM), and logistic regression (LR) for Indian origin scientists. Conversely, SMOTE with an 80-20 random split demonstrated superior performance on smaller datasets. Overall, the random forest classifier demonstrated the most favorable outcomes, attaining a micro-average area under curve (AUC) of 90%. The outcomes of our study provide a solid foundation for further investigation into classification of text of Indian origin scientists.

Corresponding Author:

Shivani Gautam
Chitkara University School of Engineering and Technology, Chitkara University
Himachal Pradesh, India
Email: shivani.gautam@chitkarauniversity.edu.in

1. INTRODUCTION

Currently, the vast majority of prominent search engines enable users to easily access information on the internet. Users primarily rely on subject-specific data for their searches. Search engines employ bots to collect data. Crawlers, often known as spiders, are responsible for traversing the internet in order to refresh and update their site material. A focused web crawler selectively indexes topic-relevant web sites that meet certain criteria, rather than obtaining information from the entire internet. The concept of focused web crawling was first introduced by Chakrabarti *et al.* [1]. A focused web crawler is designed to seek, collect, and index web pages that specifically pertain to well-defined subjects within a small portion of the internet. Since the focus is limited to a certain section of the internet, fewer network resources are used, and the required computational power is significantly reduced. This type of crawling enables the collection of pertinent and superior data. Kumar *et al.* [2], [3] proposed a method that utilizes a focused web crawler to retrieve data on academicians of Indian ancestry, based on keywords. Lunn *et al.* [4] explores the efficacy of web scraping in obtaining substantial volumes of data from web pages for dataset construction.

The current solutions are plagued by several drawbacks, such as ineffective scraping techniques, inadequate server response, and irregular data updates. On the other hand, the scraping process suggested in this study is highly proficient in retrieving data and turning it into a well-structured format with great efficiency. The objective of our research article is to improve the efficiency of the web crawler by integrating supplementary functionalities, optimized web scraping approaches, and enhanced text preprocessing methods. This will lead to a web crawler that is faster, more efficient, and more accurate.

Presently, the Indian diaspora consists of 32 million individuals, including both persons of Indian origins (PIOs) and non-resident Indians (NRIs). A diaspora population is a substantial group of individuals who have a shared geographical and cultural heritage but currently live outside their original homeland. In order to meet the rigorous criteria for a scientist's database pertaining to a certain diaspora, we have introduced the concept of targeted web crawling to gather information on scientists of Indian origin who are working in foreign nations. Data will be collected via crawling the websites of national laboratories worldwide. The compiled database of Indian origin scientists would facilitate the connection between scientists in India and their counterparts working overseas, fostering research cooperation. The research will involve examining the websites of foreign national laboratories and utilizing focused crawling techniques to create a database.

The aim of the research is to develop a targeted crawler for extracting precise information from the internet. The proposed web crawler is specifically designed to gather information about Indian-origin scientists who are employed in foreign countries. The primary goals are to: i) To systematically retrieve, categorize, and extract information about scientists of Indian origin up to a predetermined level of data retrieval; and ii) To compile a targeted lexicon of terms for guiding the web crawling process. The acquired output includes: i) A database of Indian-origin scientists employed in national laboratories abroad; and ii) A focused web crawler capable of gathering the required data.

The focused web crawler [5] utilizes vertical search engines to locate webpages that are specifically related to a particular topic, rather than collecting all web pages from the entire internet. It precisely identifies online pages that are pertinent to a particular topic and disregards irrelevant pages. Suebchua *et al.* [6] introduced a neighborhood feature that significantly enhances the effectiveness of focused web crawlers. The studies [7] and [8] offer a comprehensive analysis of different focused crawler approaches in order to optimize the harvest ratio.

The keyword query-based focused crawler employs pertinent terms associated with a particular topic to initiate queries in the search interface. The Sandhan system [9] exemplifies a focused crawler that operates based on keyword queries. It specializes in providing search results within the health and tourist industry. Multiple evaluation indicators, such as harvest ratio, recall, and precision, were employed. Tang *et al.* [10] introduces a focused web crawler that utilizes link anchor text to locate pertinent information related to the medical subject of depression. Altingövde and Ulusoy [11] presents a web crawler that utilizes subject taxonomy to determine the relevancy of web sites. In addition, the notion of tunnelling was incorporated to enhance the harvest ratio. Mukherjee [12] created a WTMS crawler to gather online pages that are tailored to a particular subject. Gatial *et al.* [13] introduces a specialized web crawler that relies on the analysis of textual content. The studies [14] and [15] provide a comprehensive analysis of many advanced targeted crawlers. Language-specific crawlers [16], [17] are implemented to enable targeted crawling of web pages published in Thai for indexing purposes. According to the information provided, Tamura *et al.* [18] utilized a focused crawler to precisely gather pages that are exclusive to a particular language. Zhao *et al.* [19] suggested SmartCrawler architecture for extracting data from the deep web. Zhang and Lu [20] presented an SCTWC methodology that enhances crawling performance by selectively choosing URLs that are relevant to the topic. Seyfi *et al.* [21] presented a Treasure crawler that used a hierarchical framework to assign a priority score to each unvisited link. Du *et al.* [22] proposed an enhanced method for improving the performance of focused crawlers by integrating term frequency-inverse document frequency (TF-IDF) values and semantic similarities. Bergmark *et al.* [23] suggested employing a tunnelling technique to improve performance in conjunction with focused crawlers for the purpose of constructing digital libraries. Goyal *et al.* [24] introduced a genetic method that relies on automated web page classification. Yan and Pan [25] proposed an augmented genetic algorithm-based focused crawler that utilizes an improved fitness function. Farag *et al.* [26] suggested a method that combines an event model with a focused crawler to efficiently retrieve relevant web sites.

Web scraping is the process of extracting unstructured material from the internet and organizing it into a structured dataset [27]. There are multiple techniques for gathering data from a webpage. Web scraping can be performed either manually by a corporation or with the use of a browser extension, application, or software. An alternative approach is the direct duplication and insertion of information from a webpage.

However, this process can become arduous when dealing with substantial amounts of data [28]. Furthermore, data can be collected instantaneously if the aforementioned website possesses its middleware. However, each provider employs different approaches to retrieve the data, which may involve a high cost for utilizing the API, and there could be certain protocols in place for accessing the information [29]. Conversely, one can directly obtain the HTML content of the web page to collect pertinent data by utilizing different computer languages such as C, Python, C++, or Node.js [30]. Once the data has been gathered, it may require additional refinement and analysis. Thus, in order to automatically obtain the data, the process of web scraping is implemented [31]. This is a bot designed to extract data from webpages. Furthermore, it has the capability to store the data in an organized format within the system for future assessments. The use of automated web scraping is necessary due to the exponential growth in the volume of data generated over time [32]. Karthikeyan *et al.* [33] introduced a model that employs efficient web scraping approaches to achieve best outcomes and accuracy. Nevertheless, there are obstacles such as inconsistent data transformation and inadequate server response. Anglin [34] outlines diverse web scraping approaches and text classification methodologies that enable scientists to filter out problematic states and districts from policy documents, allowing for the reorganization of data for evaluation purposes. Schedlbauer *et al.* [35] examines the labor market for medical informatics through the application of diverse online scraping techniques. Kaur [36] employed web scraping techniques to examine real-time news data for sentiment analysis.

The Python programming language can be employed for the entire script, in conjunction with the Beautiful Soup module for web scraping. Subsequently, the data that has been extracted can be stored in a .csv format. Python is recognized as an interpreted, object-oriented, high-level programming language [37]. Its design prioritizes optimal readability. Researchers worldwide utilize it. The programming language includes a comprehensive standard library, but it also allows for the inclusion of other libraries and toolkits to enhance its functionality. According to Khder [38], Python is the optimal programming language for implementing web scraping. Selenium is an open-source tool used to facilitate browser automation. While Selenium is mostly used for testing, it can also be employed for web scraping. It is the most commonly accepted web scraping tool and is extensively utilized with Python for automating the Chrome browser. Additionally, it has the capability to function in headless mode for web browsers. Selenium employs various techniques to locate elements, including *XPath*, *tag_name*, *class_name*, *partial_link_text*, *name*, and *css_selector*. Selenium uses the web driver to automate operations on several browsers, including Chrome, Opera, Firefox, and Microsoft Edge. Therefore, the WebDriver controller is utilized to obtain the ChromeDriver that is compatible with the current version of the browser. Manjari *et al.* [39] has explored the utilization of selenium for web scraping in order to obtain textual summaries from web sites. Han and Anderson [40] employs the usage of selenium to extract online hotel data. Bhargava *et al.* [41] employed both Selenium WebDrivers and the Beautiful Soup Library for the purpose of web scraping. Thota and Ramez [42] employed many web scraping technologies to extract news headlines and stories. Figure 1 depicts the sequential procedure employed for web scraping.

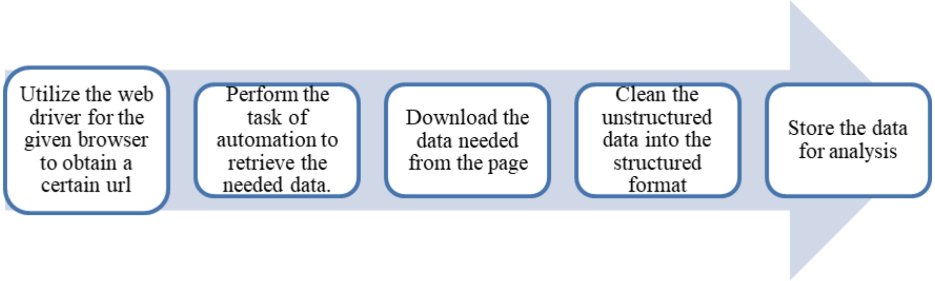

Utilize the web driver for the given browser to obtain a certain url
Perform the task of automation to retrieve the needed data.
Download the data needed from the page
Clean the unstructured data into the structured format
Store the data for analysis

Figure 1. Web scraping process

2. PROPOSED DESIGN

2.1. Design and architecture

This section pertains to the conventional design and construction of the currently employed focused crawler. Figure 2 depicts the fundamental schematic illustration for the proposed methodology. A compilation of initial URLs is created and entered into the system. The web scraping process, utilizing the *Beautiful Soup* package, retrieves all the URLs found inside the specified seed URL. Each URL in the provided list undergoes pre-processing. A keyword matching search is conducted to determine the appropriate URLs. Next, web pages that are relevant to the topic are retrieved and saved. We utilized the

Python programming language in conjunction with the Selenium framework. The relevant data is subsequently exported to a .csv file. The scientists' information is extracted and placed in the database, based on the employed data extraction procedure. Next, data mining and refinement techniques are applied to generate the final database. Furthermore, classifiers are trained and evaluated on the retrieved datasets so as to enable their use for prediction. The workflow of the suggested architecture is shown in Algorithm 1.

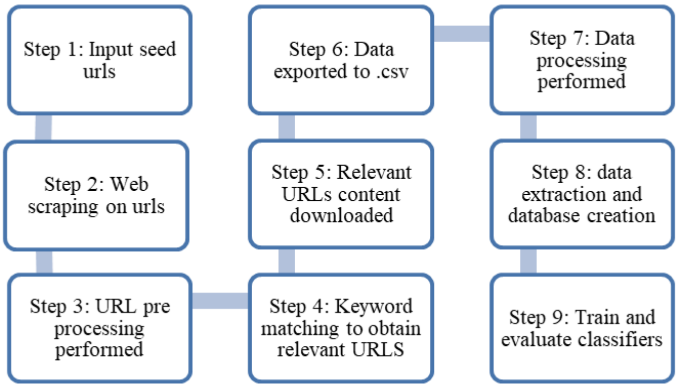

Step 1: Input seed urls
Step 2: Web scraping on urls
Step 3: URL pre processing performed
Step 4: Keyword matching to obtain relevant URLS
Step 5: Relevant URLs content downloaded
Step 6: Data exported to .csv
Step 7: Data processing performed
Step 8: data extraction and database creation
Step 9: Train and evaluate classifiers

Figure 2. Design and architecture

Algorithm 1. Workflow of the suggested architecture

```
Stage 1: URL filtering
1.  The seed URL is provided as input
2.  All the URLs found within the seed URL are obtained via web scraping
3.  For every URL in the provided list
4.  URLs undergo pre-processing, which includes tokenization, removal of stop words, and stemming
5.  If the tokens are equal to keywords like persons, staff, directory, contacts, or search, then
6.  Relevant web URLs
7.  Alternatively, irrelevant URLs
Stage 2: Data processing
1.  All the relevant URLs that have been filtered are treated as seed URLs
2.  For every applicable URL in the list
3.  Using the Selenium program, one can automate the act of downloading web pages by populating a list of surnames or designations
4.  For every webpage that is downloaded
5.  Execution of processing (tokenization, stop words removal, and stemming) takes place
6.  Create a final dataset in .csv format
7.  For every name retrieved from the .csv file
    If the fetched name or fetched university is found in the dictionary of Indian surnames and Indian universities
8.  Then the label_data is set to 1
9.  If not, assign the value of label_data as 0
10. Otherwise, the URL is not relevant
```

2.2. Crawling constraints

In order to implement the suggested architecture, several limitations on crawling are established, including the depth of crawling, seed URLs, and domain-specific keyword databases. Specialized crawlers only process seed URLs that are relevant to a given topic. Subsequently, in order to acquire seed URLs, the SeoQuake addon for Google Chrome is utilized to extract all the URLs of foreign national laboratories from a specific country. SeoQuake, a complimentary plug-in, is compatible with web browsers like Mozilla Firefox, Microsoft Edge, and Opera. Additionally, it is employed to extract the results into a .csv file for any organic search data. As a result, we obtained a list of seed URLs that may be utilized as input for our crawler to visit and extract the data.

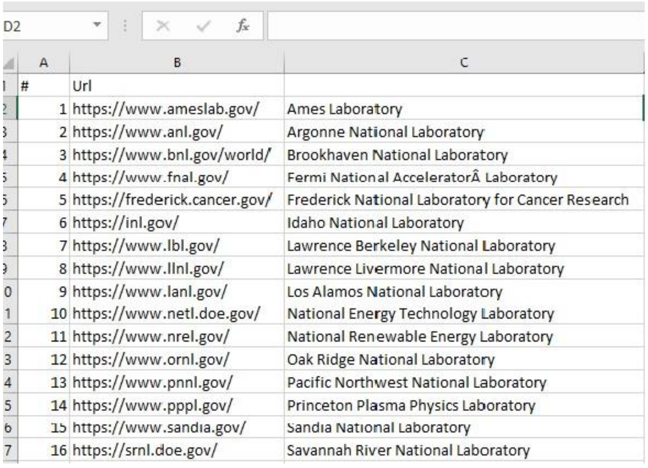
Subsequently, it is necessary to compile a database of keywords that are relevant to the domain. Keywords are employed to determine the relevance of a page. In order to collect data on Indian-origin scientists employed in foreign nations, we compile a targeted set of keywords to match against the content of downloaded web sites. This allows us to determine the relevance of each web page. The webpage is searched for designations/posts of scientists employed in foreign national laboratories, surnames of Indian individuals, and a list of Indian universities. The database stores web pages that are pertinent. If a webpage does not adhere to the dictionary of keywords, it is deemed irrelevant and the hyperlinks on that webpage are disregarded. Web pages that include terms such as people, staff, directory, contacts, and search in their URL are deemed more significant than other webpages.

Subsequently, it is necessary to determine the depth of crawling in order to collect relevant data. The attainment was achieved by setting a maximum depth, causing the crawler to halt once it reached the specified limit. A thorough search was conducted on 12 seed URLs from different international research institutes to determine the maximum depth at which relevant data was accessible. From now on, we will restrict the crawling to a maximum depth of 3, at which point the crawler should cease its operation.

3. METHOD

3.1. Acquiring initial URLs

This section thoroughly examines the implementation of the suggested architecture. The SeoQuake extension for Google Chrome is used to extract the URLs of national laboratories in a specific country. SeoQuake provided us with a list of seed URLs, which we used as input for the focused crawler to retrieve the data. A collection of over 2,000 URLs from more than 20 nations worldwide has been compiled. As depicted in Figure 3, the crawler automatically stores all the collected URLs in a .csv file for subsequent individual selection.



	A	B	C
1	#	Url	
2	1	https://www.ameslab.gov/	Ames Laboratory
3	2	https://www.anl.gov/	Argonne National Laboratory
4	3	https://www.bnl.gov/world/	Brookhaven National Laboratory
5	4	https://www.fnal.gov/	Fermi National AcceleratorÂ Laboratory
6	5	https://frederick.cancer.gov/	Frederick National Laboratory for Cancer Research
7	6	https://inl.gov/	Idaho National Laboratory
8	7	https://www.lbl.gov/	Lawrence Berkeley National Laboratory
9	8	https://www.llnl.gov/	Lawrence Livermore National Laboratory
0	9	https://www.lanl.gov/	Los Alamos National Laboratory
1	10	https://www.netl.doe.gov/	National Energy Technology Laboratory
2	11	https://www.nrel.gov/	National Renewable Energy Laboratory
3	12	https://www.ornl.gov/	Oak Ridge National Laboratory
4	13	https://www.pnnl.gov/	Pacific Northwest National Laboratory
5	14	https://www.pppl.gov/	Princeton Plasma Physics Laboratory
6	15	https://www.sandia.gov/	Sandia National Laboratory
7	16	https://srnl.doe.gov/	Savannah River National Laboratory

Figure 3. Retrieved seed URLs

3.2. URL segregation

At this stage, the initial seed URL is chosen as the input from the list of seeds. The web scraping process utilizing the *Beautiful Soup* library retrieves all the URLs found in the specified seed URL. Each URL in the obtained list undergoes pre-processing, which includes tokenization, stop words removal, and stemming. The retrieved tokens are compared with a list of terms, including persons, staff, directory, contacts, and search. If there is a successful match, the URLs are regarded relevant. Otherwise, they are considered irrelevant and removed.

The subsequent libraries and packages were employed specifically: i) *Beautiful Soup* is a Python package that allows for the extraction of data from XML and HTML files [43]; ii) The *lxml* library is a Python toolkit used for processing XML and HTML files [44]; iii) *Request* is a Python package that is utilized for creating HTTP requests; iv) *Urllib3* is a Python library that is utilized for the purpose of managing URLs; and v) *Selenium* is a Python library that is utilized for navigating through web browsers [45].

3.3. Downloading and processing of data

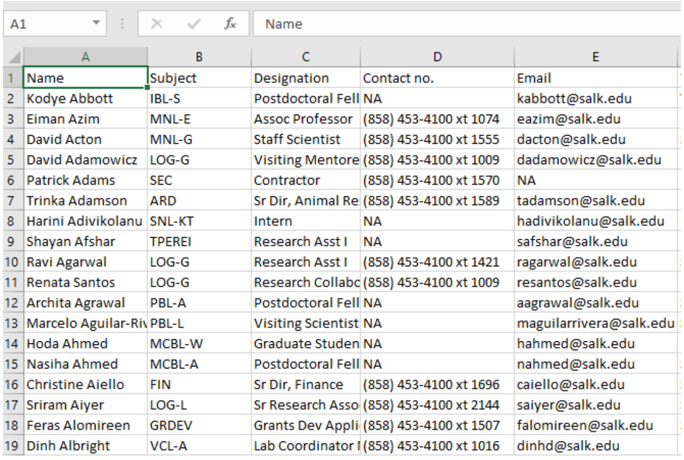
At this step, all the URLs that have been filtered and are relevant are considered seed URLs. For each relevant URL in the list, the online web pages are retrieved using the Selenium program, which automates the process of obtaining web pages by inputting various surnames or designations. The data is exported to a .csv file. Processing is conducted for each downloaded web page, encompassing the following tasks: i) The process of dividing the text into individual words is carried out, which is known as tokenization; ii) The HTML tags are eliminated; iii) Elimination of special characters such as single and double quotes, punctuation marks, etc. from the .csv file; iv) Stop words are often encountered terms that are utilized in sentence construction. Stop words in the English language encompass common terms such as "is", "the", "are", "of", "in", and "and". During the preprocessing stage, repetitious words that contribute little significance to the broader context are discarded or removed from the text; v) Stemming is the process of reducing words to their root form or stem. Stemming is also carried out at this step; vi) The process of spell checking has been completed; and vii) Abbreviations and acronyms are managed. Subsequently, a final dataset is generated in .csv format, taking the form of a list.

3.4. Indian scientists’ retrieval

In the next stage, a previously produced comprehensive set of keywords, often known as a dictionary, is referenced. This dictionary comprises two categories of keywords: Indian surnames and Indian university names. Every name extracted from the .csv file is compared and cross-checked with the dictionary collection. This comparison enables the recognition of pertinent data linked to the individual. Furthermore, the education details are cross-referenced with the roster of Indian universities to guarantee precision and pertinence. This procedure effectively retrieves and acquires the required and needed data of scientists of Indian origin.

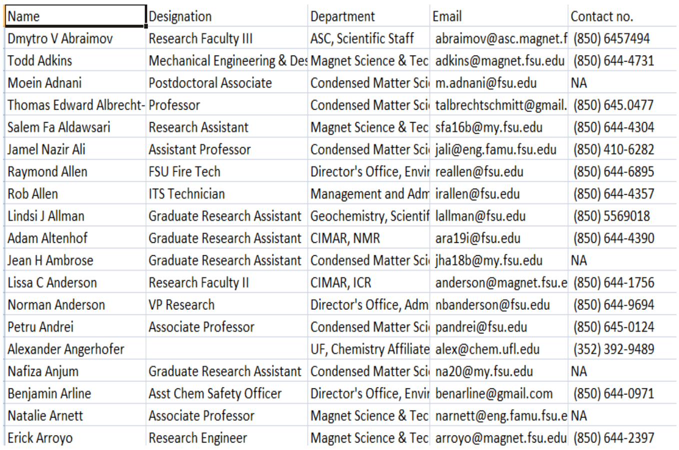
3.5. Experimental setup

The Python program utilized for the focused crawler is an experimental system that runs on a PyCharm environment on an Intel Core i5 system with a Windows 10 operating system and 8 GB of RAM. The allocated memory for the process is 1,024 MB. A Python script for web scraping is developed and installed, along with all the necessary libraries. The average internet speed typically reaches approximately 75 Mbps, while maintaining a consistent connection remains an ongoing difficulty. The issue of network disconnection frequently interrupts the crawler's data harvesting process, resulting in significant time and data loss when scraping information from a single website. The focused crawler yields promising results, generating a comprehensive database of scientists from different laboratories. This is demonstrated in Figures 4, 5, and 6, which display sample data from three distinct laboratories.



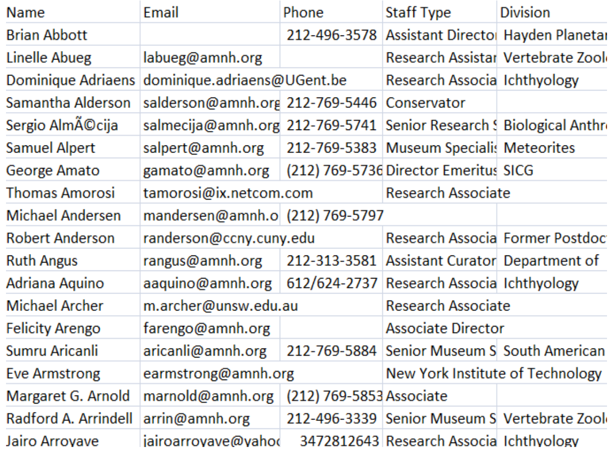
	A	B	C	D	E
1	Name	Subject	Designation	Contact no.	Email
2	Kodye Abbott	IBL-S	Postdoctoral Fell	NA	kabbott@salk.edu
3	Eiman Azim	MNL-E	Assoc Professor	(858) 453-4100 xt 1074	eazim@salk.edu
4	David Acton	MNL-G	Staff Scientist	(858) 453-4100 xt 1555	dacton@salk.edu
5	David Adamowicz	LOG-G	Visiting Mentore	(858) 453-4100 xt 1009	dadamowicz@salk.edu
6	Patrick Adams	SEC	Contractor	(858) 453-4100 xt 1570	NA
7	Trinka Adamson	ARD	Sr Dir, Animal Re	(858) 453-4100 xt 1589	tadamson@salk.edu
8	Harini Adivikolanu	SNL-KT	Intern	NA	hadivikolanu@salk.edu
9	Shayan Afshar	TPEREI	Research Asst I	NA	safshar@salk.edu
10	Ravi Agarwal	LOG-G	Research Asst I	(858) 453-4100 xt 1421	ragarwal@salk.edu
11	Renata Santos	LOG-G	Research Collabc	(858) 453-4100 xt 1009	resantos@salk.edu
12	Archita Agrawal	PBL-A	Postdoctoral Fell	NA	aagrawal@salk.edu
13	Marcelo Aguilar-Riv	PBL-L	Visiting Scientist	NA	maguilarrivera@salk.edu
14	Hoda Ahmed	MCBL-W	Graduate Studen	NA	hahmed@salk.edu
15	Nasiha Ahmed	MCBL-A	Postdoctoral Fell	NA	nahmed@salk.edu
16	Christine Aiello	FIN	Sr Dir, Finance	(858) 453-4100 xt 1696	caiello@salk.edu
17	Sriram Aiyer	LOG-L	Sr Research Asso	(858) 453-4100 xt 2144	saiyer@salk.edu
18	Feras Alomireen	GRDEV	Grants Dev Appli	(858) 453-4100 xt 1507	falomireen@salk.edu
19	Dinh Albright	VCL-A	Lab Coordinator I	(858) 453-4100 xt 1016	dinhd@salk.edu

Figure 4. Sample of .csv file (Salk) created



Name	Designation	Department	Email	Contact no.
Dmytro V Abraimov	Research Faculty III	ASC, Scientific Staff	abraimov@asc.magnet.f	(850) 6457494
Todd Adkins	Mechanical Engineering & Des	Magnet Science & Tec	adkins@magnet.fsu.edu	(850) 644-4731
Moein Adnani	Postdoctoral Associate	Condensed Matter Sci	m.adnani@fsu.edu	NA
Thomas Edward Albrecht-	Professor	Condensed Matter Sci	talbrechtschmitt@gmail.	(850) 645.0477
Salem Fa Aldawsari	Research Assistant	Magnet Science & Tec	sfa16b@my.fsu.edu	(850) 644-4304
Jamel Nazir Ali	Assistant Professor	Condensed Matter Sci	jali@eng.famu.fsu.edu	(850) 410-6282
Raymond Allen	FSU Fire Tech	Director's Office, Envir	reallen@fsu.edu	(850) 644-6895
Rob Allen	ITS Technician	Management and Adm	irallen@fsu.edu	(850) 644-4357
Lindsi J Allman	Graduate Research Assistant	Geochemistry, Scientif	lallman@fsu.edu	(850) 5569018
Adam Altenhof	Graduate Research Assistant	CIMAR, NMR	ara19i@fsu.edu	(850) 644-4390
Jean H Ambrose	Graduate Research Assistant	Condensed Matter Sci	jha18b@my.fsu.edu	NA
Lissa C Anderson	Research Faculty II	CIMAR, ICR	anderson@magnet.fsu.e	(850) 644-1756
Norman Anderson	VP Research	Director's Office, Adm	nbanderson@fsu.edu	(850) 644-9694
Petru Andrei	Associate Professor	Condensed Matter Sci	pandrei@fsu.edu	(850) 645-0124
Alexander Angerhofer		UF, Chemistry Affiliate	alex@chem.ufl.edu	(352) 392-9489
Nafiza Anjum	Graduate Research Assistant	Condensed Matter Sci	na20@my.fsu.edu	NA
Benjamin Arline	Asst Chem Safety Officer	Director's Office, Envir	benarline@gmail.com	(850) 644-0971
Natalie Arnett	Associate Professor	Magnet Science & Tec	narnett@eng.famu.fsu.e	NA
Erick Arroyo	Research Engineer	Magnet Science & Tec	arroyo@magnet.fsu.edu	(850) 644-2397

Figure 5. Sample of .csv file (National MagLab) created



Name	Email	Phone	Staff Type	Division
Brian Abbott		212-496-3578	Assistant Directo	Hayden Planeta
Linelle Abueg	labueg@amnh.org		Research Assista	Vertebrate Zool
Dominique Adriaens	dominique.adriaens@UGent.be		Research Associa	Ichthyology
Samantha Alderson	salderson@amnh.org	212-769-5446	Conservator	
Sergio AlmÃ©cija	salmecija@amnh.org	212-769-5741	Senior Research S	Biological Anthr
Samuel Alpert	salpert@amnh.org	212-769-5383	Museum Speciali	Meteorites
George Amato	gamato@amnh.org	(212) 769-5736	Director Emeritus	SICG
Thomas Amorosi	tamorosi@ix.netcom.com		Research Associate	
Michael Andersen	mandersen@amnh.o	(212) 769-5797		
Robert Anderson	randerson@ccny.cuny.edu		Research Associa	Former Postdoc
Ruth Angus	rangus@amnh.org	212-313-3581	Assistant Curator	Department of
Adriana Aquino	aaquino@amnh.org	612/624-2737	Research Associa	Ichthyology
Michael Archer	m.archer@unsw.edu.au		Research Associate	
Felicity Arengo	farengo@amnh.org		Associate Director	
Sumru Aricanli	aricanli@amnh.org	212-769-5884	Senior Museum S	South American
Eve Armstrong	earmstrong@amnh.org		New York Institute of Technology	
Margaret G. Arnold	marnold@amnh.org	(212) 769-5853	Associate	
Radford A. Arrindell	arrin@amnh.org	212-496-3339	Senior Museum S	Vertebrate Zool
Jairo Arroyave	jairoarroyave@yahoo	3472812643	Research Associa	Ichthyology

Figure 6. Sample of .csv file (AMNH) created

3.6. Data analysis

Upon concluding the crawling procedure and obtaining the requisite data, the subsequent step is analyzing the acquired data. The crawler is implemented with diverse parameter values and depths to capture the data and assess the efficacy and constraints of our code. Subsequently, the gathered data is analyzed to ascertain the anticipated precision of the data's depth, as well as the necessary bandwidth for each website. This analysis aids in determining a suitable indexing technique for effective information retrieval. We also perform an exploratory analysis of the datasets to obtain valuable insights that could have an impact on the text categorization challenge. Nevertheless, a primary consideration in the development of a classification model is the balance of the distinct classes. This entails ensuring that the dataset contains a roughly equal distribution of each label. The obtained files provide an illustration of the distribution of the gathered dataset. The three datasets namely National Museum of Natural History (AMNH), National MagLab and Salk contain 583, 712, and 1,116 records respectively. Out of these, our analysis shows that 80% of the scientists are non-Indians, while only 20% are Indians. Hence, it is evident that our dataset exhibits an imbalance. This can exemplify the accuracy paradox, where our accuracy is high, yet we still make inaccurate predictions for all classes. Hence, we used other metrics to assess the effectiveness of our model, as accuracy is an inadequate statistic for imbalanced data sets. Addressing the problem of class imbalance is a critical undertaking in the domain of machine learning, particularly when working with datasets that exhibit a significant discrepancy in the representation of several classes. Imbalanced data refers to a dataset in which the distribution of observations in the target class is disproportionate. One effective approach for addressing class imbalance is the utilization of synthetic minority oversampling technique (SMOTE). The SMOTE algorithm operates by producing artificial samples for the underrepresented class, so augmenting its presence within the dataset.

3.7. Evaluation metrics

Evaluation metrics are closely associated with tasks in the domain of machine learning. There exist numerous methodologies for evaluating the performance of classification algorithms. Accuracy, recall, precision, F1-score are widely recognized as prominent evaluation metrics in several domains [46].

– Accuracy: Accuracy is the ratio of accurate forecasts to the total number of predictions. The effectiveness of the method is contingent upon the presence of an equal distribution of samples across all classes.
– Recall: Recall denotes the ratio of accurately identified positive outcomes to the total number of pertinent samples.
– Precision: Precision is the proportion of correctly predicted positive results out of the total number of positive results expected by the classifier.
– F1-score: The F1-score is the arithmetic mean of precision and recall, determined using the harmonic mean.

4. RESULTS AND DISCUSSION

In this section, we have performed many experiments to assess the efficacy of different models in the text categorization approach. Afterwards, we classify our collected datasets by utilizing several supervised machine learning techniques. The Random Forest algorithm with cross validation demonstrated the highest level of performance, with a micro-average AUC of 90% for the similar domain [47]. Moving further, three distinct datasets were gathered, specifically the AMNH, National MagLab, and Salk databases. We have employed two methodologies to conduct comparative analysis on these three datasets. In the initial method, we employed SMOTE and executed a random division, assigning 80% of the observations for training data and 20% for the test data set. The second strategy involved employing the SMOTE technique and doing hyper parameter tuning with cross-validation on the training data to enhance the model's accuracy. Here, we display the output of our experiments with various models used for classification of text on Indian origin scientist's dataset. We evaluated the performance of our models by employing certain measures such as precision, accuracy, recall, and F1-score. The measurements offer useful insights into the model's efficacy in managing multi-class situations [48], [49]. When evaluating the models and selecting the optimal hyper parameters, we considered accuracy. We also utilized the classification report to calculate precision, recall, and F1-score. The methods employed include random forest (RF), K-nearest neighbors (KNN), support vector machine (SVM), decision tree (DT), simple cart (SC), and logistic regression (LR). The outcomes of our studies are presented in Figure 7 show casing a comparison of metrics such as accuracy, precision, recall, and F1-score for the first strategy utilizing random division with 80-20 split. The findings of our study suggest that none of the six learning algorithms consistently demonstrate strong performance. However, the random forest classifier exhibits the highest overall performance. More results presented in Figures 8 and 9.

Figure 10 presents a comparison of metrics such as accuracy, precision, recall, and F1-score for the second strategy utilizing k-fold cross validation. Our findings suggest that among the six learning algorithms, the random forest classifier demonstrates the highest overall performance. More results presented in Figure 11 and Figure 12.

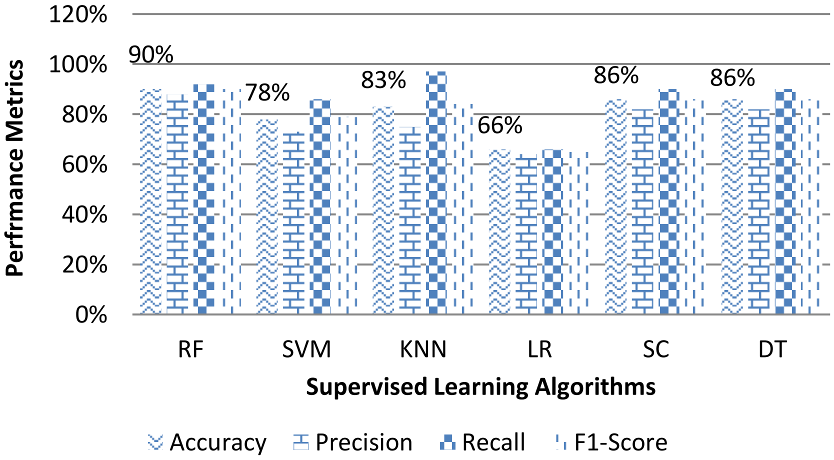


Figure 7. Classification results (performance metrics for 80-20 random split) AMNH dataset

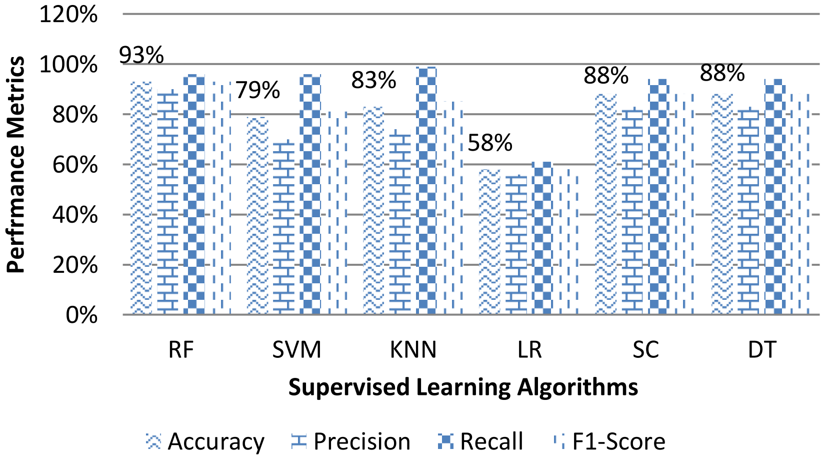


Figure 8. Classification results (performance metrics for 80-20 random split) National MagLab dataset

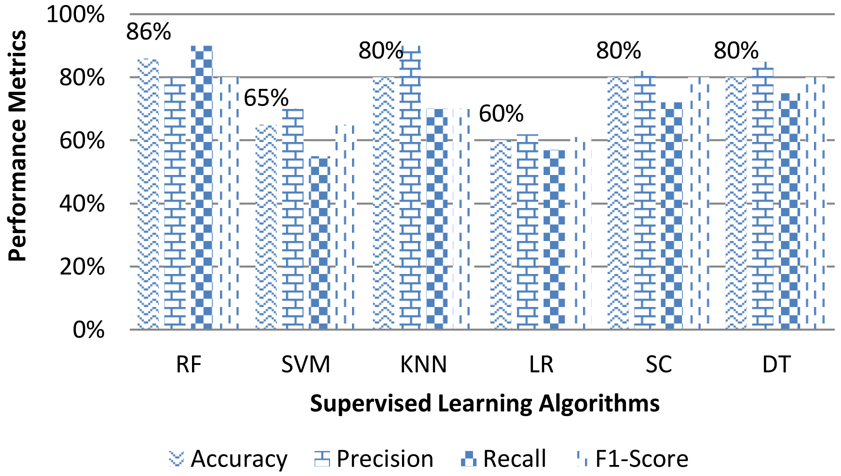


Figure 9. Classification results (performance metrics for 80-20 random split) Salk dataset

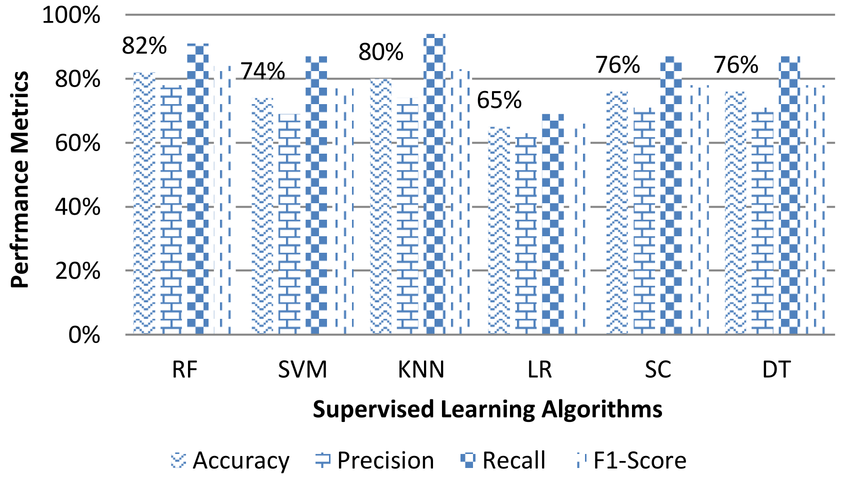


Figure 10. Classification results (performance metrics for 10 fold cross validation) AMNH dataset

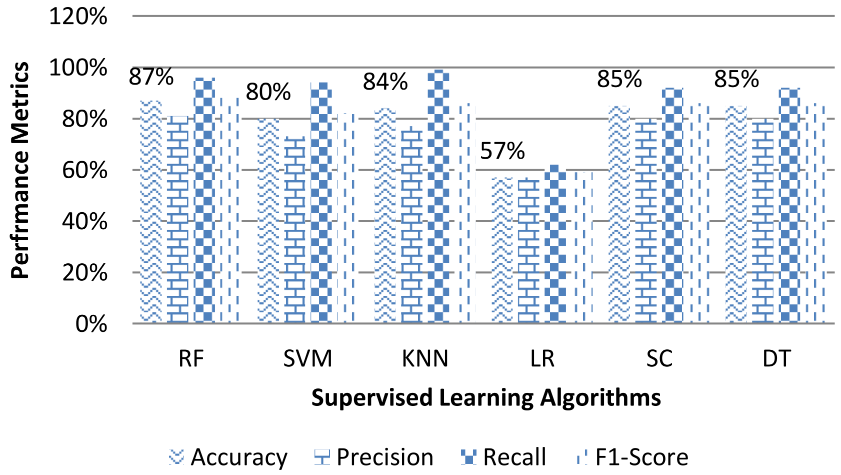


Figure 11. Classification results (performance metrics for 10 fold cross validation) National MagLab dataset

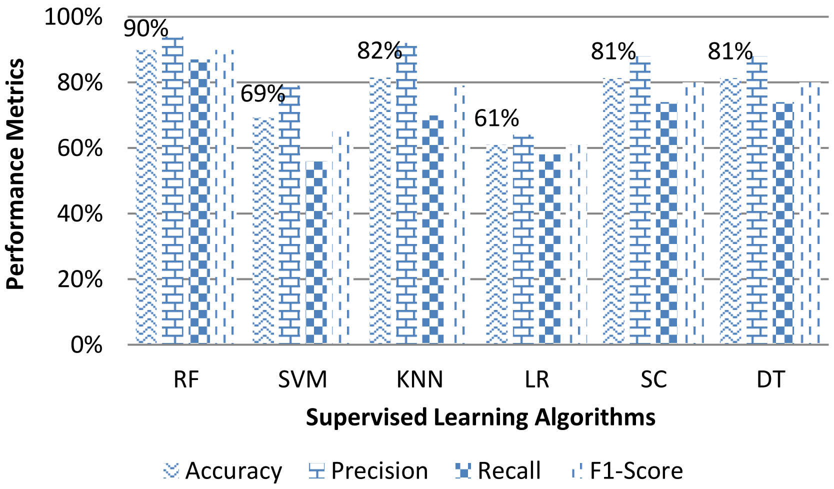


Figure 12. Classification results (performance metrics for 10 fold cross validation) Salk dataset

We have implemented two approaches to conduct comparative analysis. In the initial approach, we applied the SMOTE technique and employed a random split, 80% of the data will be used for training purposes, while the remaining 20% will be allocated for testing. In the second method, we employed the SMOTE technique and performed hyper parameter optimization using cross-validation on the training data to train the model. When evaluating the results of the two techniques, specifically the 80-20 random split and k-fold cross validation, it is evident that the 80-20 random split outperforms the 10-fold cross validation for small datasets such as AMNH and National MagLab. This is demonstrated by achieving accuracies of 90% and 93% respectively, as shown in Figure 7 and Figure 8. K-fold cross-validation demonstrates enhanced performance when applied to larger datasets such as Salk, as illustrated in Figure 12, achieving a 90% accuracy rate.

This analysis demonstrates the efficacy of the 80-20 random split method for smaller datasets and the benefits of using k-fold cross-validation for larger datasets. Various interpretations can be derived from this discovery, each providing valuable understanding of the features and consequences of these methods. Furthermore, it is crucial to take into account the constraints of the investigation. Smaller datasets such as AMNH and National MagLab tend to derive greater advantages from an 80-20 random split. This approach offers a clear division of the data, guaranteeing that the model is trained on a substantial percentage of the data while still retaining enough for testing. Equilibrium is essential for smaller datasets in which each data point holds substantial importance. K-fold cross-validation is particularly advantageous for larger datasets like Salk because it guarantees that each data point is utilized for both training and testing, resulting in a thorough assessment of the model. It aids in reducing the variability and systematic error that may occur due to a single division of data into training and testing sets.

In smaller datasets, the likelihood of overfitting is increased because there is a limited amount of data available for training. The 80-20 split is used to maintain a balance between training and testing data, which helps to reduce overfitting and improve generalization. K-fold cross-validation is a reliable method for evaluating larger datasets as it calculates the performance by averaging across numerous folds. This methodology facilitates the capture of the fluctuations in the data, resulting in a more dependable assessment of model performance and enhanced generalization.

There are a few limitations as well. The conclusions are derived from distinct datasets (AMNH, National MagLab, Salk), and the outcomes may differ when using alternative datasets. The attributes of these datasets, such as the distribution of features and the balance of classes, can impact the efficacy of the techniques. Although the study indicates overall patterns, the extent to which these findings may be applied to other situations or areas may be restricted. Distinct domains may display distinct data characteristics that can impact the efficacy of the 80-20 split or k-fold cross-validation.

Ultimately, the 80-20 random split seems to be more efficient for smaller datasets, whereas k-fold cross-validation is more suitable for larger datasets. However, it is important to consider the limits of the study and the individual characteristics of the datasets when interpreting these conclusions. In order to fully generalize these discoveries, it is crucial to do additional study and testing using a wide range of datasets.

5. CONCLUSION

This study emphasizes the need of data retrieval for Indian scientists working in outside National laboratories. Manually searching through such a vast amount of material is an unattainable endeavor. Hence, it is necessary to develop a focused crawler capable of traversing websites through the utilization of web scraping methodologies. The implemented system would facilitate global scientific collaboration by enabling scientists around the world from India to communicate with their counterparts in India. This research employs efficient web scraping methodologies to initially extract data from websites, which is subsequently transformed into an organized manner. The data is prepared and optimized for future procedures using natural language processing (NLP) techniques. The significance of a webpage is determined by a keyword matching algorithm. The effectiveness of the crawling process depends on the depth of the webpage, which acts as the endpoint. By restricting the crawler to URLs within the domain of the parent seed URL, it can achieve precise and targeted results. Consequently, the created crawler has the ability to autonomously navigate websites and retrieve data. Furthermore, we utilized the text classification techniques on retrieved datasets for comparative analysis. The task of text classification entails employing supervised machine learning methods to train a model using preprocessed input. Throughout our investigations, we evaluated the effectiveness of various supervised models. The evaluation results demonstrate that the SMOTE with Standard Random Forest model, utilizing 10-fold cross validation, outperformed other models and achieved the highest F1-score of 90% when handling large datasets. Conversely, the SMOTE with standard random forest model using an 80-20 random split displayed superior performance in smaller datasets. Overall, the Random Forest classifier demonstrated the most favorable outcomes, attaining a micro-average AUC of 90%. The results of the aforementioned investigation illustrate the optimal method for text classification.

In our future endeavors, we plan to enhance the effectiveness of the models by refining them on a larger dataset. Additionally, we aim to expand the research to include a wider range of natural language processing approaches. In addition, we can determine the temporal efficiency of our method in future endeavors. Furthermore, we have future intentions to categorize the data of scientists of Indian descent utilizing unsupervised algorithms. Subsequently, we will juxtapose their findings with those of the supervised models. In our future endeavors, our goal is to create a comprehensive dataset of Indian-origin physicians who are currently practicing in medical facilities located outside of India.

REFERENCES

[1] S. Chakrabarti, M. Van Den Berg, and B. Dom, “Focused crawling: a new approach to topic-specific Web resource discovery,” *Computer Networks*, vol. 31, no. 11, pp. 1623–1640, 1999, doi: 10.1016/S1389-1286(99)00052-3.

[2] M. Kumar, R. Bhatia, A. Ohri, and A. Kohli, “Design of focused crawler for information retrieval of Indian origin Academicians,” in *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, Apr. 2016, pp. 1–6, doi: 10.1109/ICACCA.2016.7578895.

[3] M. Kumar, A. Bindal, R. Gautam, and R. Bhatia, “Keyword query based focused Web crawler,” *Procedia Computer Science*, vol. 125, pp. 584–590, 2018, doi: 10.1016/j.procs.2017.12.075.

[4] S. Lunn, J. Zhu, and M. Ross, “Utilizing web scraping and natural language processing to better inform Pedagogical practice,” in *2020 IEEE Frontiers in Education Conference (FIE)*, Oct. 2020, pp. 1–9, doi: 10.1109/FIE44824.2020.9274270.

[5] M. Shokouhi, P. Chubak, and Z. Raeesy, “Enhancing focused crawling with genetic algorithms,” in *International Conference on Information Technology: Coding and Computing, ITCC*, 2005, vol. 2, pp. 503–508, doi: 10.1109/itcc.2005.145.

[6] T. Suebchua, B. Manaskasemsak, A. Rungsawang, and H. Yamana, “Efficient topical focused crawling through neighborhood feature,” *New Generation Computing*, vol. 36, no. 2, pp. 95–118, 2018, doi: 10.1007/s00354-017-0029-8.

[7] I. Avraam and I. Anagnostopoulos, “A comparison over focused web crawling strategies,” in *2011 15th Panhellenic Conference on Informatics*, Sep. 2011, pp. 245–249, doi: 10.1109/PCI.2011.53.

[8] S. Batsakis, E. G. M. Petrakis, and E. Milios, “Improving the performance of focused web crawlers,” *Data and Knowledge Engineering*, vol. 68, no. 10, pp. 1001–1013, 2009, doi: 10.1016/j.datak.2009.04.002.

[9] P. N. Priyatam, S. R. Vaddepally, and V. Varma, “Proceedings of the first workshop on information and knowledge management for developing region,” in *Proceedings of the First Workshop on Information and Knowledge Management for Developing Region*, Nov. 2012, pp. 23–30, doi: 10.1145/2389776.2389782.

[10] T. T. Tang, D. Hawking, N. Craswell, and K. Griffiths, “Focused crawling for both topical relevance and quality of medical information,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, Oct. 2005, pp. 147–154, doi: 10.1145/1099554.1099583.

[11] I. S. Altingövde and Ö. Ulusoy, “Exploiting interclass rules for focused crawling,” *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 66–73, 2004, doi: 10.1109/MIS.2004.62.

[12] S. Mukherjea, “WTMS: a system for collecting and analyzing topic-specific Web information,” *Computer Networks*, vol. 33, no. 1, pp. 457–471, 2000, doi: 10.1016/S1389-1286(00)00035-9.

[13] E. Gatial, Z. Balogh, M. Laclavík, M. Ciglan, and L. Hluchý, “Focused web crawling mechanism based on page relevance,” in *ITAT 2005 - Workshop on Theory and Practice of Information Technologies - Applications and Theory, Proceedings*, 2005, pp. 41–46.

[14] F. Menczer, G. Pant, and P. Srinivasan, “Topical web crawlers: evaluating adaptive algorithms,” *ACM Transactions on Internet Technology*, vol. 4, no. 4, pp. 378–419, 2004, doi: 10.1145/1031114.1031117.

[15] M. Kumar, R. Bhatia, and D. Rattan, “A survey of web crawlers for information retrieval,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 6, 2017, doi: 10.1002/widm.1218.

[16] P. Tadapak, T. Suebchua, and A. Rungsawang, “A machine learning based language specific web site crawler,” in *2010 13th*

International Conference on Network-Based Information Systems, Sep. 2010, pp. 155–161, doi: 10.1109/NBiS.2010.25.

[17] E. Srisukha, S. Jinarat, C. Hamechaiyasak, and A. Rungsawang, "Naïve bayes based language-specific web crawling," in *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2008*, 2008, vol. 1, pp. 113–116, doi: 10.1109/ECTICON.2008.4600385.

[18] T. Tamura, K. Somboonviwat, and M. Kitsuregawa, "A method for language-specific Web crawling and its evaluation," *Systems and Computers in Japan*, vol. 38, no. 2, pp. 10–20, 2007, doi: 10.1002/scj.20693.

[19] F. Zhao, J. Zhou, C. Nie, H. Huang, and H. Jin, "SmartCrawler: a two-stage crawler for efficiently harvesting deep-web interfaces," *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 608–620, 2016, doi: 10.1109/TSC.2015.2414931.

[20] H. Zhang and J. Lu, "SCTWC: an online semi-supervised clustering approach to topical web crawlers," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 490–495, 2010, doi: 10.1016/j.asoc.2009.08.017.

[21] A. Seyfi, A. Patel, and J. Celestino Júnior, "Empirical evaluation of the link and content-based focused treasure-crawler," *Computer Standards and Interfaces*, vol. 44, pp. 54–62, 2016, doi: 10.1016/j.csi.2015.09.007.

[22] Y. Du, W. Liu, X. Lv, and G. Peng, "An improved focused crawler based on semantic similarity vector space model," *Applied Soft Computing Journal*, vol. 36, pp. 392–407, 2015, doi: 10.1016/j.asoc.2015.07.026.

[23] D. Bergmark, C. Lagoze, and A. Sbityakov, "Focused crawls, tunneling, and digital libraries," in *Research and Advanced Technology for Digital Libraries: 6th European Conference*, 2002, pp. 91–106.

[24] N. Goyal, R. Bhatia, and M. Kumar, "A genetic algorithm based focused web crawler for automatic webpage classification," in *IET Conference Publications*, 2016, vol. 2016, pp. 1–6, doi: 10.1049/cp.2016.1546.

[25] W. Yan and L. Pan, "Designing focused crawler based on improved genetic algorithm," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, Mar. 2018, pp. 319–323, doi: 10.1109/ICACI.2018.8377476.

[26] M. M. G. Farag, S. Lee, and E. A. Fox, "Focused crawler for events," *International Journal on Digital Libraries*, vol. 19, no. 1, pp. 3–19, 2018, doi: 10.1007/s00799-016-0207-1.

[27] S. de S. Sirisuriya, "A comparative study on web scraping," in *Proceedings of 8th International Research Conference*, 2015, pp. 135–140.

[28] S. M. Iacus, "Automated data collection with R - A practical guide to web scraping and text mining," *Journal of Statistical Software*, vol. 68, no. Book Review 3, pp. 1–452, 2015, doi: 10.18637/jss.v068.b03.

[29] S. R. Mani Sekhar, G. M. Siddesh, S. S. Manvi, and K. G. Srinivasa, "Optimized focused web crawler with natural language processing based relevance measure in bioinformatics web sources," *Cybernetics and Information Technologies*, vol. 19, no. 2, pp. 146–158, 2019, doi: 10.2478/cait-2019-0021.

[30] J. Ward, *Instant PHP web scraping*. Packt Publishing, 2013.

[31] E. Vargiu and M. Urru, "Exploiting web scraping in a collaborative filtering-based approach to web advertising," *Artificial Intelligence Research*, vol. 2, no. 1, pp. 44–54, 2012, doi: 10.5430/air.v2n1p44.

[32] A. Sinha, M. N. B. J. Naskar, M. Pandey, and S. S. Rautaray, "Text classification using machine learning techniques: comparative analysis," in *2022 OITS International Conference on Information Technology (OCIT)*, Dec. 2022, vol. 4, no. 8, pp. 102–107, doi: 10.1109/OCIT56763.2022.00029.

[33] T. Karthikeyan, K. Sekaran, D. Ranjith, V. Vinoth Kumar, and J. M. Balajee, "Personalized content extraction and text classification using effective web scraping techniques," *International Journal of Web Portals*, vol. 11, no. 2, pp. 41–52, 2019, doi: 10.4018/IJWP.2019070103.

[34] K. L. Anglin, "Gather-narrow-extract: a framework for studying local policy variation using web-scraping and natural language processing," *Journal of Research on Educational Effectiveness*, vol. 12, no. 4, pp. 685–706, 2019, doi: 10.1080/19345747.2019.1654576.

[35] J. Schedlbauer, G. Raptis, and B. Ludwig, "Medical informatics labor market analysis using web crawling, web scraping, and text mining," *International Journal of Medical Informatics*, vol. 150, 2021, doi: 10.1016/j.ijmedinf.2021.104453.

[36] P. Kaur, "Sentiment analysis using web scraping for live news data with machine learning algorithms," *Materials Today: Proceedings*, vol. 65, pp. 3333–3341, 2022, doi: 10.1016/j.matpr.2022.05.409.

[37] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit 1st edition, kindle edition*. O'Reilly Media, Inc., 2009.

[38] M. A. Khder, "Web scraping or web crawling: State of art, techniques, approaches and application," *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 3, pp. 144–168, 2021, doi: 10.15849/ijasca.211128.11.

[39] K. U. Manjari, S. Rousha, D. Sumanth, and J. Sirisha Devi, "Extractive text summarization from web pages using selenium and TF-IDF algorithm," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, Jun. 2020, pp. 648–652, doi: 10.1109/ICOEI48184.2020.9142938.

[40] S. Han and C. K. Anderson, "Web scraping for hospitality research: overview, opportunities, and implications," *Cornell Hospitality Quarterly*, vol. 62, no. 1, pp. 89–104, 2021, doi: 10.1177/1938965520973587.

[41] R. Bhargava, R. Lobo, R. Shah, N. Shah, and S. Nair, "Easier web navigation using intent classification, web scraping and NLP approaches," in *5th IEEE International Conference on Advances in Science and Technology, ICAST 2022*, 2022, pp. 286–290, doi: 10.1109/ICAST55766.2022.10039559.

[42] P. Thota and E. Ramez, "Web scraping of COVID-19 news stories to create datasets for sentiment and emotion analysis," in *Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference*, Jun. 2021, pp. 306–314, doi: 10.1145/3453892.3461333.

[43] L. Richardson, "Beautiful soup," *crummy.com*, 2018. https://www.crummy.com/software/BeautifulSoup/ (accessed Jan. 20, 2024).

[44] Scoder, "lxml - XML and HTML with Python" *GitHub*, 2024, https://github.com/lxml/lxml (accessed Dec. 05, 2023).

[45] S. Nyamathulla, P. Ratnababu, N. Sultana Shaik, B. N. Lakshmi, and A. Professor, "A review on selenium web driver with Python," *Annals of R.S.C.B.*, vol. 25, no. 4, pp. 16760–16768, 2021.

[46] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Lecture Notes in Networks and Systems*, 2023, vol. 724, pp. 15–25, doi: 10.1007/978-3-031-35314-7_2.

[47] S. Gautam, R. Bhatia, and S. Jain, "Classification and analysis for focused crawled textual dataset for retrieving Indian origin scientists," *International Journal of Experimental Research and Review*, vol. 34, pp. 72–85, 2023, doi: 10.52756/ijerr.2023.v34spl.008.

[48] W. Xing and A. Ghorbani, "Weighted PageRank algorithm," in *Proceedings - Second Annual Conference on Communication Networks and Services Research*, 2004, pp. 305–314, doi: 10.1109/dnsr.2004.1344743.

[49] T. Suebchua, A. Rungsawang, and H. Yamana, "Adaptive focused website segment crawler," in *NBiS 2016 - 19th International Conference on Network-Based Information Systems*, 2016, pp. 181–187, doi: 10.1109/NBiS.2016.5.

BIOGRAPHIES OF AUTHORS

Shivani Gautam received the MCA degree from Kurukshetra University, India, and currently pursuing Ph.D. degree from Chitkara University, Himachal Pradesh. Currently, she is an assistant professor at the Department of Computer Science and Engineering, Chitkara University, Himachal Pradesh. Her areas of interest are machine learning, focused crawlers, artificial intelligence, and information retrieval. She has a total of 14 years of experience in teaching. She can be contacted at email: shivani.gautam@chitkarauniversity.edu.in.

Rajesh Bhatia is a professor in Department of Computer Science and Engineering, PEC University of Technology, Chandigarh, INDIA. He has PhD in computer science. He has more than 25 years of experience in teaching. His areas of interest are software testing, software clone detection, software engineering, software component retrieval, information retrieval. He has authored more than 100 papers. Currently he is dean, Academic Affairs, Punjab Engineering College, India. His Ph.D. students are working in the area of automated software debugging, semantic software clone detection and automated test case generation, information retrieval, focused web crawling. He can be contacted at email: rbhatiapatiala@gmail.com.

Shaily Jain is PhD in computer science in the year 2014. Prior to this she had her master’s in technology in 2009. She has a total of 18 years of experience in teaching. Her research interests include multiprocessor systems on chip, embedded systems, wireless networks, security in networks, data mining and education engineering. She has in total 40+ international journals and conference papers in her credits. She has guided 10 MTech research students and 5 PhD students. Her current affiliation is professor, CSE in Chitkara University, Punjab, India. She can be contacted at email: shaily.jain@chitkarauniversity.edu.in.