Extracting Academic Publication Data Using Web Automation Tools

Zurina Saaya
Fakulti Teknologi Maklumant dan Komunikasi
Universiti Teknikal Malaysia Melaka
Email: zurina@utem.edu.my

Abstract— Web automation is an approach to programmatically accessing websites to perform several tasks such as software testing, web scraping, forms completion or transferring data between websites. In this paper, we describe the development of a web automation tools to extract academic publication data from online bibliographic database namely Google Scholar and Scopus for an academic institution. The extracted data is then will be analyzed to produce publication reports to measure scholarly productivity and impact by academic staff, departments, institutions and research groups. With this information, the top management can monitor publication performance in a better way in term of decision making and organizational strategic planning. In addition, the reports can be used to emphasize and promote individual staff and grant applications from certain research group or department based of publication output.

Index Terms— web automation, scraping, publication data, data extraction.

I. Introduction

Academic staff at any institute is assigned to a number of responsibilities such as teaching, research, supervision, publication and consultancy. In some condition research and publication may get a higher point on productivity than other responsibilities when evaluating them for appointment and promotion for certain position. Over time, awarding tenure and/or academic promotion has been linked to the publications and the citation of those publications [1].

One of the important measurements related to publication data is the number of publications by an academic staff. The other measurement is citation analysis. These measurements can be used to assess publication impact based on number of subsequent publications cite a publication [2][3][4].

H-index is another important measure which is beyond simple citation counts. The h-index is an index that is used to measure the productivity and impact of the publications of an author. Nowadays, h-index is increasingly being used to evaluate performance of an author (academic staff). This measurement also important for funding agencies in comparing academic institutions performance. Beyond these measurements, the method of assessing author impact from a single perspective is lack in certain condition, for example how to measure performance for several authors in one research group.

Currently, online bibliographic database and publication search engine are useful for searching publication records for academic staff. However, they do not provide feature for extracting and reporting tools at institution level. This tool is required to summarize articles and publication performance for particular entity such as academic staff, research group, department or institute. Tool with reporting functionality may help users to easily understand the data and make effective decision, such as which research domain are having better future directions for academic staff or research group. Whether to put more or reduce funding by the research management center or even provide prospects for investment by industries to invest in a research project.

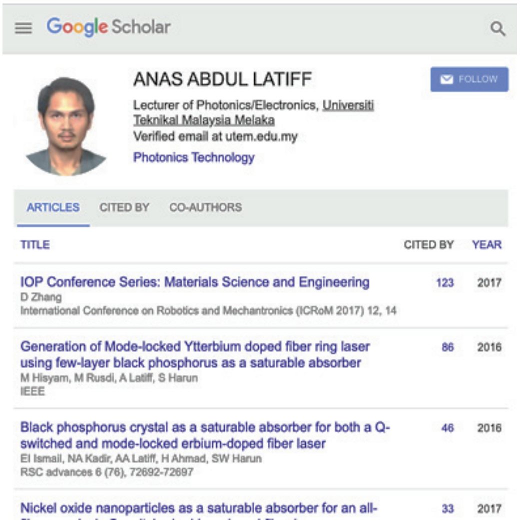
In this paper, we explain the development of a web automation tool to extract academic publication data from online bibliographic database and search engine namely Google Scholar and Scopus, then analyze these data to produce publication reports for a specific staff, research group, department or institute. This web automation tool is a part of the component in UTeMAIR crawling engine [5].

By employing this automation tool, universities can now monitor and update their academic publication information automatically to keep track the performance of their academic staff particularly for their research and publication assessment. Besides that, this information can be utilized to develop statistical analysis and keywords analysis module to organize information through abstractions and aggregation. This can benefit university management i.e. research center unit finding trends and important information regarding staff publication. With the availability of this information, the university can monitor scholarly information in a better way and plan towards increasing the publication index among academics and ultimately improve the university academic impact.

II. Related work

Mostly, academic publication is published as journal article, book or thesis and are indexed by offline or online bibliographic database. Google Scholar is an example for online bibliographic database. It indexes the full text or metadata of scholarly publication from numerous digital libraries, different types of publishing formats and disciplines. These databases and search tools basically provide an organized list of articles based on search query entered by user [6]. If the query in the form of article title, the result will show list of articles related to the given query. Besides that, Google Scholar have feature to view author profile page as in Fig. 1 and Fig. 2. Each author that has been recognized by Google Scholar will have their own profile page. As we can see in these figures, we can get author's affiliation, research area, list of articles, year of publication, list of co-authors, citation for each article and citation analysis for individual author.

Scopus is another example of general bibliographic database that indexes journal articles, conference proceedings and books. Besides searching tool, Scopus also provides features to analyze and visualize research topics for specific author or institute as depicted in Fig. 3. As the data is in Scopus database it is difficult to customize the report based on institutional needs. For example, to summarize publication performance for certain academic designations or identify prominent research topics among research group members.



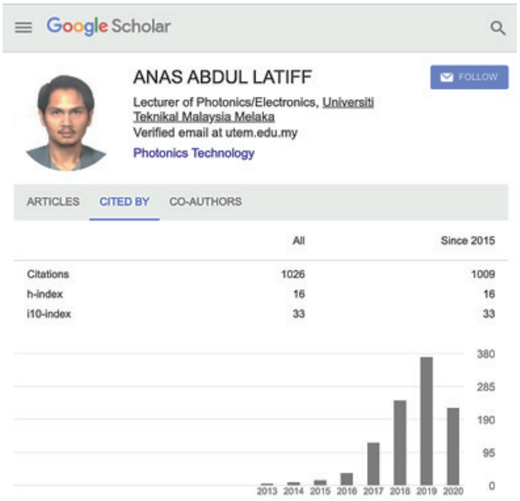
ANAS ABDUL LATIFF

Lecturer of Photonics/Electronics, Universiti Teknikal Malaysia Melaka
Verified email at utem.edu.my
Photonics Technology

ARTICLES CITED BY CO-AUTHORS

TITLE	CITED BY	YEAR
IOP Conference Series: Materials Science and Engineering D Zhang International Conference on Robotics and Mechantronics (ICRoM 2017) 12, 14	123	2017
Generation of Mode-locked Ytterbium doped fiber ring laser using few-layer black phosphorus as a saturable absorber M Hisyam, M Rusdi, A Latiff, S Harun IEEE	86	2016
Black phosphorus crystal as a saturable absorber for both a Q-switched and mode-locked erbium-doped fiber laser El Ismail, NA Kadir, AA Latiff, H Ahmad, SW Harun RSC advances 6 (76), 72692-72697	46	2016
Nickel oxide nanoparticles as a saturable absorber for an all-	33	2017

Fig. 1. Google Scholar author’s profile page.



ANAS ABDUL LATIFF

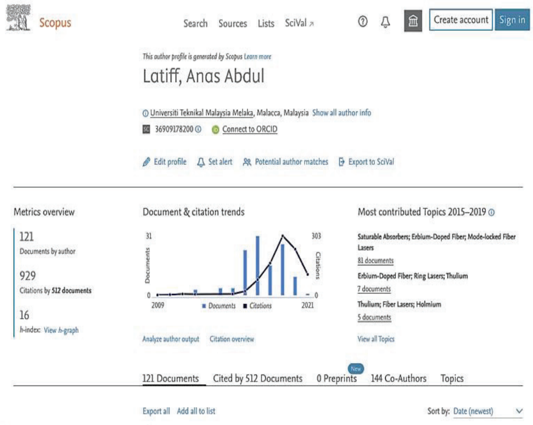
Lecturer of Photonics/Electronics, Universiti Teknikal Malaysia Melaka
Verified email at utem.edu.my
Photonics Technology

ARTICLES CITED BY CO-AUTHORS

	All	Since 2015
Citations	1026	1009
h-index	16	16
i10-index	33	33

Fig. 2. Google Scholar author’s citation analysis

Besides general bibliographic database, there are also domain-specific databases for example, IEEE Xplore and ACM Digital Library in Computer Science and Informatics area. For biomedical topics and life sciences the most popular database is PubMed. Most online bibliographic database provides search engine to search for publication and return the result consists of relevant of articles, however there is no sufficient features to explore the search results for example the summarization the abstracts for a number of related articles and research areas. Most of the time, the search result is filtered and sorted based on relevancy, author name, author affiliations, publication date, article keywords and citation scores.



This author profile is generated by Scopus Learn more

Latiff, Anas Abdul

Universiti Teknikal Malaysia Melaka, Malacca, Malaysia Show all author info
36909178200 Connect to ORCID

Metrics overview

121
Documents by author

929
Citations by 512 documents

16
h-index: View h-graph

Document & citation trends

Most contributed Topics 2015–2019

Saturable Absorbers; Erbium-Doped Fiber; Mode-locked Fiber Lasers
81 documents
Erbium-Doped Fiber; Ring Lasers; Thulium
7 documents
Thulium; Fiber Lasers; Holmium
5 documents
View all Topics

Fig. 3. Scopus author’s profile page

Web automation primarily used for software testing as automation tools to reduce human intervention and repeatable tasks. Automation testing able to automate the testing process or procedures including design and execution of test scripts. By using automation testing the quality of software testing has been improved and able to be simplifed since it can minimize human intervention [7].

Beside software testing, web automation also can be utilized and customized to programmatically accessing websites to perform other tasks such as web scraping, completing forms, or transferring data between websites. There are several tools that have been developed for this purpose namely, Ringer [8], Rousillon [9] and WebCombine [10]. These tools records user’s activities and uses it as input, then it will produce automation script that interacts with the page based on recorded user’s activities. Since these tools derive the automation scripts by recoding user demonstration or activities, it is an advantage for non-programmers to use web automation for to fulfill their purpose. However, it is lack of ability for customization especially when there is a need to manipulate the automation process. As for this paper the approach of programming by demonstration is not suitable since there are specific inputs are required (i.e a list user’s information) and the web data that is extracted from the resources need to be kept in a specific database.

For developing our own web automation tools, we will adapt web automation based on the script written in advance. Since the objective of this tool already been set so there is no need to create a recorded script. Besides, the development process is much simpler than automation by demonstration. Our system is implemented based on Selenium WebDriver [11][12] as the main automation tools for data extraction. Web automation using Selenium test-suite family has been widely adapted in software development lifecycle particularly is testing phase specifically for automation testing. Besides automation testing Selenium also can be used in web crawling [13] and web scraping [14][15].

There many more libraries or application programming interface (API) that can be used to automate the extraction process. Some of the examples are BeautifulSoup, Scrapy, and Requests [16]. However, these libraries not applicable for dynamic pages. Beside Selenium, the other option that we can use is Puppeteer, a Node.js library that provides API to control Chrome or Chromium [17][18]. Selenium and Puppeteer are the web automation that can can operate by driving web browsers. This type of API provides interface for web browser to interact and manipulate the document object model (DOM).

III. Web automation for data extraction

In this research have developed a web automation tools to extract the academic publication data from Google Scholar and Scopus based a given list of authors. The extracted data is then will be analyzed to produce publication reports. Our web extraction tool consists of four main components as depicted in Fig. 4.

This web automation is implemented using Selenium WebDriver. Selenium WebDriver is an API that enable browser automation to accept instructions to control and automate browser activities. It is implemented through browser-specific driver such as ChromeDriver for Chrome browser. Altenatively it also can be implemented using GeckoDriver for Firefox, OperaDriver for Opera or WebDriver for Microsoft Edge browsers. It controls browser based on pre-defined automation commands. Initialy, it is created for automating web-based application testing to verify that all the functions for web application are performed as it supposed [6]. However, we utilize the capability of this tool to achieve our objectives. We have implemented this browser automation framework to scrape web data from online bibliographic database (see Fig. 3).

Communication between WebDriver and Python script is in JSON wire protocol. Basically, every statement in the automation script will be transformed into URL with the help of JSON Wire Protocol over HTTP. WebDriver uses a HTTP server to receive HTTP requests. When the URL reaches the WebDriver, the request will be passed to the browser over HTTP. Then the instructions in automation script will be executed on the browser. Once the page is successfully loaded, WebDriver will extract HTML script from the browser and sent to scraping script to be parsed then update the Publication Database.

To access all important data for each author, we need to create an automation script to automate browsing activities, for example, open each author profile page, scroll down the page, maximize the browser, and open certain hyperlinks. These activities are needed as some of the website content is loaded dynamically via JavaScript code which is activated based on user's interaction with the page. Because of this reason, common web extraction API for Python cannot be used as the data that are retrived from the server dynamically generated. Therefore, it is not able to be extracted by BeautifulSoup or Scrapy Python API.

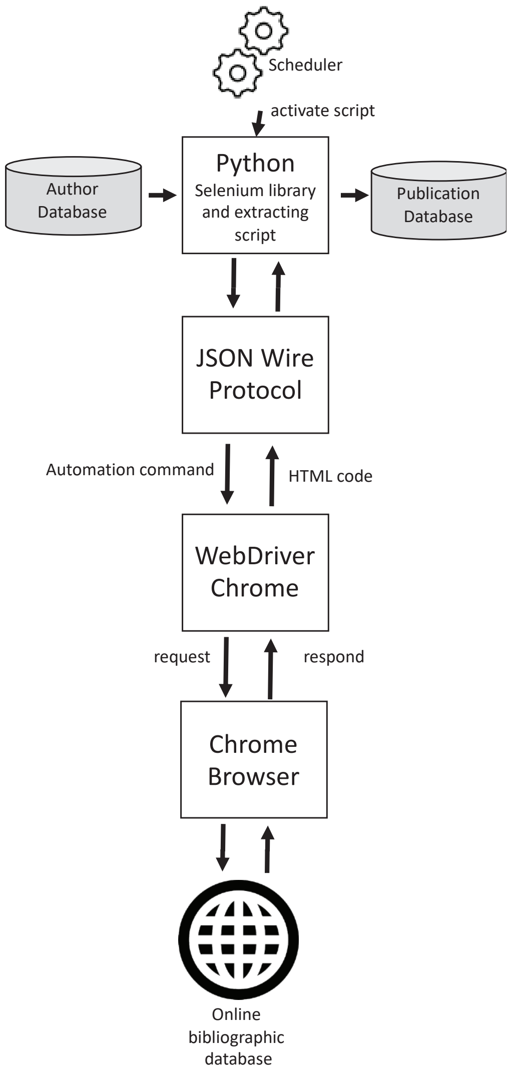


Fig. 4. System overview

System flow chart for the process of extraction the data is shown in Fig. 5, automation script will retrieve the citation page for each author based on author id which is used as seed keyword. For each page, the script will download its content to gather the details of publication data for each author. The script will perform pre-processing on the retrieved pages to clean the extracted content and perform analysis to identify unique data to avoid duplications.

For each author, the are various information can be extracted. However, we choose only to keep author's list of published articles, citation data and h-index. The data is then saved in local repository and we named as Publication database. The data from Publication database is then can be used to generate publication report for individual staff, faculty, designation and research group or. As shown in Fig. 6 - 9. The data also can be analyzed further for report generation and data visualization as in Fig. 10

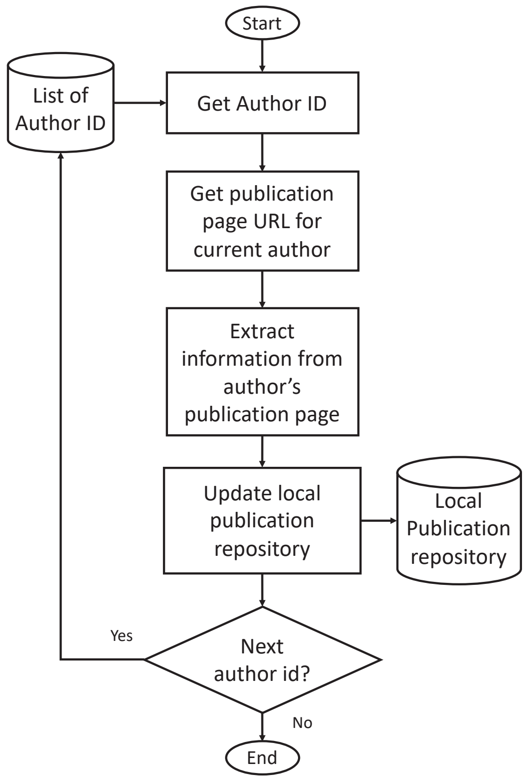
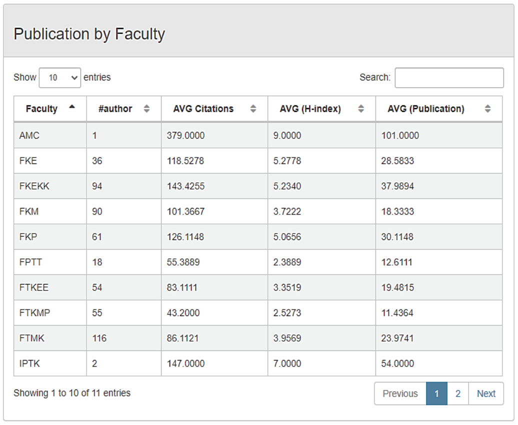

Start
List of Author ID
Get Author ID
Get publication page URL for current author
Extract information from author's publication page
Update local publication repository
Local Publication repository
Next author id?
Yes
No
End

Fig. 5. System flow chart



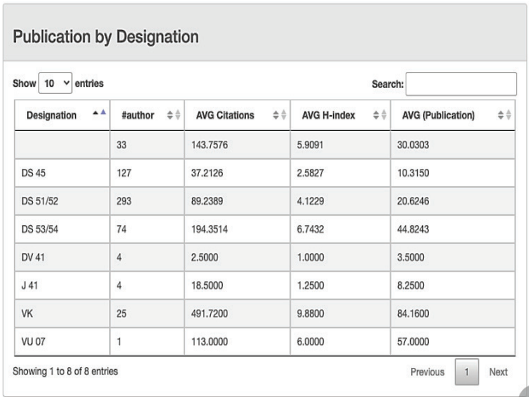
Publication by Faculty

Show 10 entries Search:

Faculty	#author	AVG Citations	AVG (H-index)	AVG (Publication)
AMC	1	379.0000	9.0000	101.0000
FKE	36	118.5278	5.2778	28.5833
FKEKK	94	143.4255	5.2340	37.9894
FKM	90	101.3667	3.7222	18.3333
FKP	61	126.1148	5.0656	30.1148
FPTT	18	55.3889	2.3889	12.6111
FTKEE	54	83.1111	3.3519	19.4815
FTKMP	55	43.2000	2.5273	11.4364
FTMK	116	86.1121	3.9569	23.9741
IPTK	2	147.0000	7.0000	54.0000

Showing 1 to 10 of 11 entries Previous 1 2 Next

Fig. 7. Sample publication data for faculty



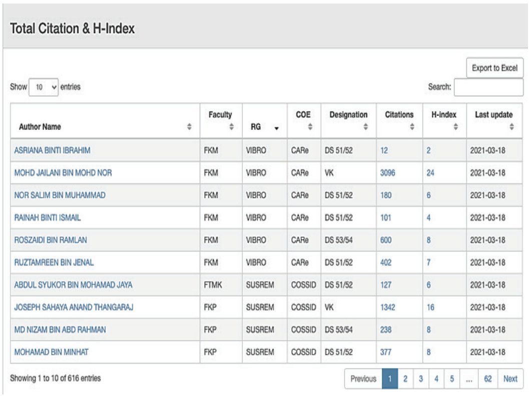
Publication by Designation

Show 10 entries Search:

Designation	#author	AVG Citations	AVG H-index	AVG (Publication)
	33	143.7576	5.9091	30.0303
DS 45	127	37.2126	2.5827	10.3150
DS 51/52	293	89.2389	4.1229	20.6246
DS 53/54	74	194.3514	6.7432	44.8243
DV 41	4	2.5000	1.0000	3.5000
J 41	4	18.5000	1.2500	8.2500
VK	25	491.7200	9.8800	84.1600
VU 07	1	113.0000	6.0000	57.0000

Showing 1 to 8 of 8 entries Previous 1 Next

Fig. 8. Sample publication data based on designation



Total Citation & H-Index

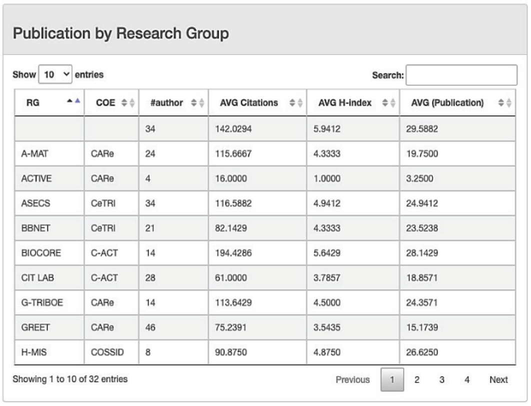
Export to Excel

Show 10 entries Search:

Author Name	Faculty	RG	COE	Designation	Citations	H-index	Last update
ASRIANA BINTI IBRAHIM	FKM	VIBRO	CARe	DS 51/52	12	2	2021-03-18
MOHD JAILANI BIN MOHD NOR	FKM	VIBRO	CARe	VK	3096	24	2021-03-18
NOR SALIM BIN MUHAMMAD	FKM	VIBRO	CARe	DS 51/52	180	6	2021-03-18
RAINAH BINTI ISMAIL	FKM	VIBRO	CARe	DS 51/52	101	4	2021-03-18
ROSZAIDI BIN RAMLAN	FKM	VIBRO	CARe	DS 53/54	600	8	2021-03-18
RUZTAMREEN BIN JENAL	FKM	VIBRO	CARe	DS 51/52	402	7	2021-03-18
ABDUL SYUKOR BIN MOHAMAD JAYA	FTMK	SUSREM	COSSID	DS 51/52	127	6	2021-03-18
JOSEPH SAHAYA ANAND THANGARAJ	FKP	SUSREM	COSSID	VK	1342	16	2021-03-18
MD NIZAM BIN ABD RAHMAN	FKP	SUSREM	COSSID	DS 53/54	238	8	2021-03-18
MOHAMAD BIN MINHAT	FKP	SUSREM	COSSID	DS 51/52	377	8	2021-03-18

Showing 1 to 10 of 616 entries Previous 1 2 3 4 5 ... 62 Next

Fig. 6. Sample publication data for each staff



Publication by Research Group

Show 10 entries Search:

RG	COE	#author	AVG Citations	AVG H-index	AVG (Publication)
		34	142.0294	5.9412	29.5882
A-MAT	CARe	24	115.6667	4.3333	19.7500
ACTIVE	CARe	4	16.0000	1.0000	3.2500
ASECS	CeTRI	34	116.5882	4.9412	24.9412
BBNET	CeTRI	21	82.1429	4.3333	23.5238
BIOCORE	C-ACT	14	194.4286	5.6429	28.1429
CIT LAB	C-ACT	28	61.0000	3.7857	18.8571
G-TRIBOE	CARe	14	113.6429	4.5000	24.3571
GREET	CARe	46	75.2391	3.5435	15.1739
H-MIS	COSSID	8	90.8750	4.8750	26.6250

Showing 1 to 10 of 32 entries Previous 1 2 3 4 Next

Fig. 9. Sample publication data based on research group

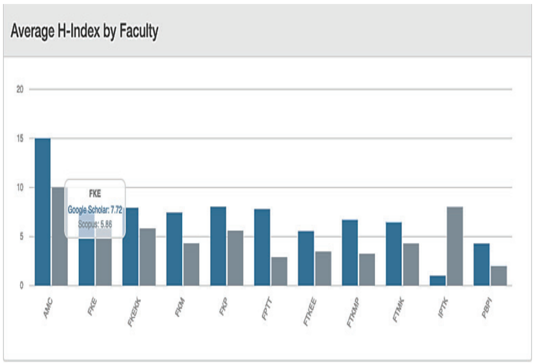


Fig. 10. Sample publication data based on research group

IV. Evaluation and Results Extraction

To evaluate the performance of the automation tools we compare our automation tool which is developed using Selenium and the other automation tool developed using Puppeteer. There two main component that will be compared are (1) time taken for the task to be completed and (2) the resources (processor time, memory) used to complete the tasks. We have created two simple script to scrape data for 100 authors (profiles) from Google Scholar using Selenium and Puppeteer.

This experiment is done on the Windows 10 machine and resources parameters are captured using Windows' Performance Monitor (Perfmon). Perfmon is a utility tool that allows users to capture and monitor various parameters of system resources. Perfmon is included with Windows operating system. User can monitor performance data either in real time or from a log file. Basically, to setup the experiment, user need to create Data Collector Sets to configure and schedule performance counter, event trace and configure data collection so that it can be analyzed and viewed in a report.

During the execution of each script, we capture three parameters (1) Duration of execution, (2) percentage Processor Time and (3) Working Set. Percentage of Processor Time shows the amount of CPU usage that a process (script) is using while Working Set indicate the amount of physical memory a process is using. On average the duration for Selenium script to finish the scraping process is 12 minutes while Puppeteer took about 14.5 minutes. In term of processor time Puppeteer used less processor resource compare to Selenium. However, for memory usage, Puppeteer use more the Selenium. Table 1 shows the average result for these three parameters.

TABLE I
RESULT SUMMARY

Parameter	Selenium	Puppeteer
Duration	12 minutes	14.5 minutes
Average % Processor Time	30%	1%
Average Working Set	25 MB	50 MB

Fig. 11 shows the details percentage of processor time used during the execution. Meanwhile, Fig. 12 shows the Working Set (memory usage) during scraping processes. These results are the sample of 50 data (timestamps) from Performance Monitor. Both show results for both Selenium and Puppeteer. For Processor time, Puppeteer use just around 1% of processor time while Selenium is in various values with maximum almost reach 70%.

The result from this experiment shows that scraping performance for Selenium and Puppeteer not much different. There are advantages and disadvantages for each approach. Therefore, the selection of tool for scraping task either Selenium or Puppeteer is depending on the developer on how they want to conduct the data collection.

There are trading-off between computational overhead, data accuracy, flexibility and developer effort need to be considered. For example, developer effort in term complexity of programming Selenium is much simpler and more direct to develop compare to Puppeteer. For flexibility, web automation with Selenium is develop using Python, therefore the tool can be enhanced and utilized all Python libraries.

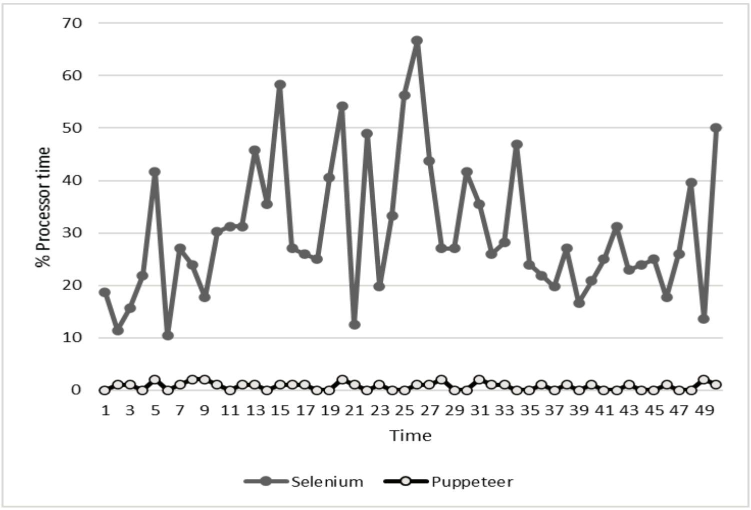


Fig. 11. Processor time

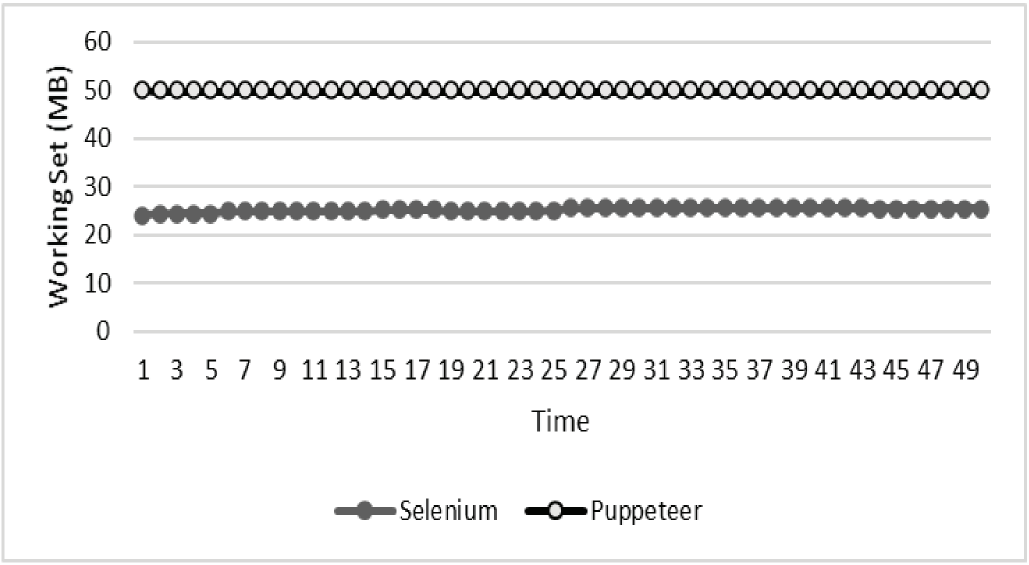


Fig. 12. Working set (memory usage)

IV. Conclusion

We have explained the development of web automation tool to extract academic publication data. The extracted data is then used to produce publication reports to measure scholarly productivity and impact by academic staff, departments, institutions, or research groups. The publication data were extracted from two popular online publication repository sites i.e. Google Scholar and Scopus. In designing this tool, our main goal is to collect publication data for a list of authors affiliated to an academic institute. To do so, we need a tool scrape the data from the web and save it in our own database. Web automation tools are very useful for scraping task. The initial purpose of web automation tool is for software testing, however with its automation feature, it is possible for data collection process specifically in web scraping. There are many options available for the developer to choose from. Since not all options are created equal or offer the same functionalities, and therefore the choice of option is critical.

Regarding scraping issues, to avoid any denial of service by the web server, the automation tool needs to be is set on a sleep mode for a random n second. This approach is significant as the standards and ethics in web scraping are met [19]. The "politeness" of the scraping tool has also been ensured to avoid blacklisting of the current IP address that could cause future problems to the users in the network. However, this automation took must be scalable towards the growth in the number of academic staff. It must be able to retrieve a large amount of information as the number of staff increases. Therefore, future enhancement needs to be done to handle this issue.

As for the future work, beside improvement in the automation tool, we hope to develop an expert search engine based on author's publication data and academic impact analytics dashboard specifically for academics intitution.

Acknowledgement

The authors would like to thank to Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia (UTeM), Centre for Advanced Computing Technology (CACT) and Computational Intelligence and Technologies (CIT) research group for their incredible supports in this project.

References

[1] Zhang, Fuli, Xiaomei Bai, and Ivan Lee. "Author Impact: Evaluations, predictions, and challenges." IEEE Access 7 (2019): 38657-38669.

[2] Garfield, Eugene. "The history and meaning of the journal impact factor." jama 295.1 (2006): 90-93.

[3] Pan, Raj Kumar, and Santo Fortunato. "Author Impact Factor: tracking the dynamics of individual scientific impact." Scientific reports 4.1 (2014): 1-7.

[4] Farooq, Muhammad, et al. "DS-index: Ranking authors distinctively in an academic network." IEEE Access 5 (2017): 19588-19596.

[5] L. B. H. Zurina Saaya Yogan Jaya Kumar, "Collecting and Analyzing Academic Publication with UTeMAIR," International Journal of Recent Technology and Engineering, vol. 8, no. 4, pp. 2357–2363, Nov. 2019, doi: 10.35940/ijrte.d8271.118419.

[6] C. Dunne, B. Shneiderman, R. Gove, J. Klavans, and B. Dorr, "Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization," Journal of the American Society for Information Science and Technology, vol. 63, no. 12, pp. 2351–2369, Nov. 2012, doi: 10.1002/asi.22652

[7] Gojare, Satish, Rahul Joshi, and Dhanashree Gaigaware. "Analysis and design of selenium webdriver automation testing framework." Procedia Computer Science 50 (2015): 341-346.

[8] S. Barman, S. Chasins, R. Bodik, and S. Gulwani, "Ringer: web automation by demonstration," ACM SIGPLAN Notices, vol. 51, no. 10, pp. 748–764, Dec. 2016, doi: 10.1145/3022671.2984020.

[9] S. E. Chasins, M. Mueller, and R. Bodik, "Rousillon: Scraping Distributed Hierarchical Web Data," In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, pp. 963-975. 2018. 2018, doi: 10.1145/ 3242587.3242661.

[10] S. Chasins, S. Barman, R. Bodik, and S. Gulwani, "Browser Record and Replay as a Building Block for End-User Web Automation Tools," In Proceedings of the 24th International Conference on World Wide Web, pp. 179-182. 2015, doi: 10.1145/2740908.2742849.

[11] Mustika, Nurul Rita. "Automated Black Box Testing using Selenium Python." International Journal of Computer Science and Software Engineering 7.9, 2018, 201-204.

[12] E. Vila, G. Novakova, and D. Todorova, "Automation Testing Framework for Web Applications with Selenium WebDriver," In Proceedings of the International Conference on Advances in Image Processing, pp. 144-150. 2017, doi: 10.1145/3133264.3133300.

[13] J. Peng, Y. Ma, F. Zhou, S. Wang, Z. Zheng, and J. Li, "Web Crawler of Power Grid Based on Selenium," In 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, pp. 114-118. IEEE, 2019.

[14] R. S. Chaulagain, S. Pandey, S. R. Basnet, and S. Shakya, "Cloud Based Web Scraping for Big Data Applications," In 2017 IEEE International Conference on Smart Cloud (SmartCloud), pp. 138-143. IEEE, 2017. Nov. 2017, doi: 10.1109/smartcloud.2017.28.

[15] K. D. Gorro, M. J. G. Sabellano, K. Gorro, C. Maderazo, and K. Capao, "Classification of Cyberbullying in Facebook Using Selenium and SVM," In 2018 3rd International Conference on Computer and Communication Systems (ICCCS), pp. 183-186. IEEE, 2018.

[16] Thomas, David Mathew, and Sandeep Mathur. "Data analysis by web scraping using python." 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2019.

[17] Guzmán Castillo, Paola, Pau Arce Vila, and Juan Carlos Guerri Cebollada. "Automatic QoE evaluation of DASH streaming using ITU-T Standard P. 1203 and Google Puppeteer." Proceedings of the 16th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks. 2019.

[18] Sarker, Shaown, Jordan Jueckstock, and Alexandros Kapravelos. "Hiding in Plain Site: Detecting JavaScript Obfuscation through Concealed Browser API Usage." Proceedings of the ACM Internet Measurement Conference. 2020. 0.

[19] Rennie, S., Buchbinder, M., Juengst, E., Brinkley-Rubinstein, L., Blue, C., & Rosen, D. L. (2020). Scraping the Web for Public Health Gains: Ethical Considerations from a 'Big Data'Research Project on HIV and Incarceration. Public Health Ethics, 13(1), 111-121.

Zurina Saaya is currently serving as a senior lecturer at Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM) where she involved with teaching computer networking topics. She has been teaching for almost 15 years. She completed her PhD in Computer Science and Informatics from Universiy College Dublin, Ireland in 2014. During her service Her research focuses on technologies for information retrieval, data mining and recommender systems. She has published more than 10 publication topics related to information retrieval, recommender system and sentiment analysis.