An Improving Approach for Fast Web Scrapping Using Machine Learning and Selenium Automation

Sunny Mehta, Prof. Gayatri Pandi (Jain)

***Abstract*— Generally, Selenium Automation is used for testing purpose and detecting errors and defects of the system in development. Here, we will use Selenium for making a list of required web element counts from web page by using Machine Learning technique Count Vectorizer and using that list selenium will identify the new documents from web pages for scrapping data. For example, if we consider any tenders site, there may be thousands of tenders getting published every day, so it is very hard for the user to surf every tender one after another. But, in our method we are using class labels or a unique id for gathering data in the first round of scrapping by using Forward Selection Wrapper methods of Feature Selection in Machine Learning to identify the new documents from web pages. In our system to be implemented web pages will automatically scrap new documents from websites by using web client or stream writer. The system implemented by us will perform faster web scrapping by using selenium automation framework and various machine learning techniques used which will be helpful for many users for less time consuming and save manual efforts in browsing data from web pages.**

***Index Terms*— Automated Testing, Selenium Web Driver, Machine Learning, Web Client, Stream Writer, Unique Id, Count Vectorizer, Forward Selection Wrapper Method.**

I. INTRODUCTION

Selenium is a portable framework for testing web applications. It also provides a **Selenese** - a test domain specific language to write tests in a number of popular programming languages, including C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala [10]. The tests can then run on most web browsers. Selenium runs on Windows, Linux, and macOS. It is open source software released under the Apache License 2.0 [16].

Figure 1: Modern browsers use to run tests of Selenium Framework.

***Sunny Mehta**, Computer Engineering, L.J. Institute of Engineering and Technology, Ahmedabad, India.*
***Prof. Gayatri Pandi (Jain)**, HOD Post Graduation Computer Department, L.J. Institute of Engineering and Technology, Ahmedabad, India.*

Web Scraping is a method to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table. Instead of manually download documents one after another - a very tedious job which can take many hours or sometimes days to complete. Web Scraping is the technique of automating this process, so that instead of downloading the documents from websites, the Web Scraping software will perform the same task within a fraction of the time [15].

To scrap or download the documents from web pages, web client or stream writer will be needed. The Web Client class provides common methods for sending data to or receiving data from any local, intranet, or Internet resource identified by a URL. The Web Client class uses the Web Request class to provide access to resources. [19]

Stream writer is used to write data to a file using streams. The data from application is written into the stream first. After that the stream will write the data to a file. [20] Figure 2 shows how stream writer works.

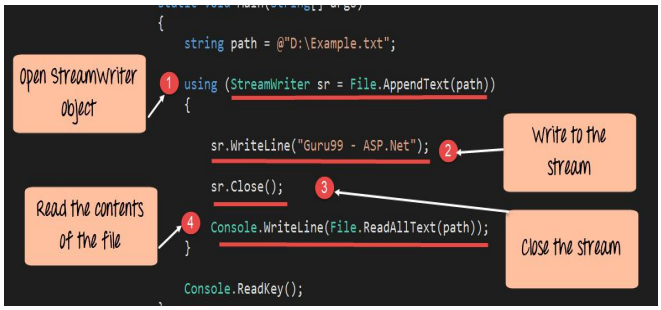

Figure 2: Example of how stream writer works.

Here, Unique Id will be nothing more than a label of the element which is unique in the web page which can be helpful for scrapping data from the second round of scrapping where the already scrapped data will be skipped and the fresh data will only be scrapped. For example, if we take tenders site then there will be many elements present but there will be one unique element in the document which we can use for scrapping purpose which is tender number which is always unique in the webpage and which will only be helpful for fresh document scrapping.

II. Related Work

Generally, in recent trends in selenium is for the better automated testing purpose and for efficient test results rather than doing a manual testing which is more time consuming and human effort is needed.

Web pages would be trained in such a perspective that system is able to understand what test cases are needed to be followed for elements on that web pages and next time on wards system will do testing on those elements on its own [1], [2], [12].

System automatically learns what actions to be done where on its own. For example, if there is a button it will click there or else if there is a text box then it will enter the required text etc. The quality of web application is one of important factor while deploying the web applications [4], [6]. So, for increase in quality of software testing plays a very important role.

By using Selenium automation framework test scripts system will automatically finds the mentioned elements from the web pages where the action is mentioned and whenever the element is not found then it will inform the user that this element is not found in the web page so it will be easy for the user to just change the script for that element and the test script will run automatically from the next time[3], [5], [9].

In some cases, testers generate some test scripts from which they can generate csv files of the test data where they can see which element have what class as well as what is it id or else what will be it type or which web element is it [1] as shown in Figure 3. Here in figure 3, it is shown that what type of the element is there in the web page with its class, tag name, type and id [13], [14]. Such test data are used in machine learning also where machine learning is for predicting test cases for each web element [1] and then selenium performing those test cases. The following figure 3 shows that type of test data where every web element will be bifurcated with its class, id, type as well as its tag name.

	A	B	C	D	E	F
1		class	id	tag_name	type	web_element
2	0	no	username	input	username	textbox
3	1	no	pass	input	password	textbox
4	2	no	noteditable	input	text	textbox
5	3	no	fname	input	text	textbox
6	4	no	lname	input	text	textbox
7	5	no	mail	input	text	textbox
8	6	no	hidden	input	hidden	textbox
9	7	no	searcharea	input	search	textbox
10	8	no	submit	input	submit	button

Figure 3: Test Data CSV File.

III. Proposed System

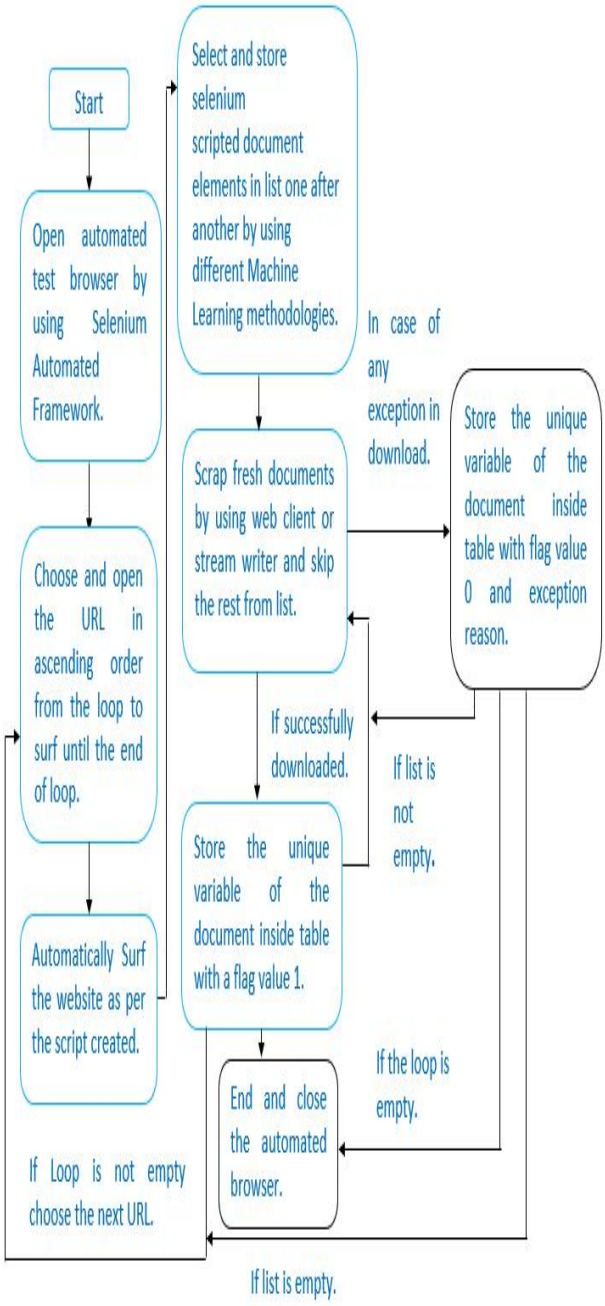

Figure 4: Flowchart of Proposed System.

A. How System Works

Figure 4 illustrates the working principles of this paper proposed system. The system starts by opening the automated test browser using selenium automation framework [7], [8], [11]. The modern browser which supports and run the test scripts of selenium are mentioned in Figure 1.

The URL needed for scrapping can be a single URL or a multiple URL also so that we will take a loop of URL where single as well as multiple URL are to be set for scrapping. Then in the next step, from the following loop of URL one after another URL are chosen and browsed in the browser by

selenium test scripts. This loop will be run in ascending order until the last URL is browsed.

Now, the surfing will be started of the URL in the next step as scripted in the selenium test script. While surfing the web pages, as per selenium test scripts documents will be selected and stored in the list by using machine learning methodologies.

Now the selected and the stored documents will be scrapped from the list one after another. In scrapping, if the scrap of that particular web page is running for first time then all the documents of the list will be downloaded or else from the second round only the fresh documents will be downloaded and the rest will be skipped from the list. By this the user will get the fresh documents only on daily basis. The system will be trained in such a manner that from second round of scrapping it will only select and download fresh documents only.

There may be the cases where many elements have the exceptions by default in the web pages. For example, there may be some elements in the web pages which are not working properly as they are scripted which means some links won't open where the directory of that file is not present or else some documents have the elements but not the links to open them. So, while scrapping these types of documents will create an exception so this system will gather these exceptions with the elements unique variable and add the flag value 0 there in the database. While if the document is successfully downloaded then also it will store the unique variable in the database with the flag value 1.

This whole process will be continuously work unit the list goes empty it will choose another URL from the loop and this whole process will be repeated for that URL also. System will choose URL one after another from the loop until the loop gets empty and once the loop gets empty it will end the process by closing the automated test browser.

B. *Machine Learning Usage in System*

In this system, Machine Learning will have a major role for extracting features from the dataset so for that Count Vectorizer is used which involves counting number of occurrences each word appears in the document [17]. So, by using Count Vectorizer system will obtain the number of the elements present in the web pages whose element was being accounted for counting.

Forward Selection Wrapper methods of Feature Selection are used in this system where we start the system with having no features in the model. In each iteration, features are added which best improves our model till an addition of a new variable does not improve the performance of the model [18].

IV. Implementation and Results

Here for fast scrapping the big bunch of documents, the site should also be that much huge so let us a take a example of some tender site where thousands of tender documents get published on daily basis so it is very hard for the user to surf every tender one after another because it will be much time consuming. But by implementing this system all the fresh documents are just scrapped and downloaded in just couple of time and it will save a lot much of time of user.

A. *Opening an Automated web browser*

Open an automated test browser by selecting the first URL from the loop of URL as scripted in selenium test script. To browse the URL in Selenium can be done by the following script as shown in the figure 5 and figure 6. After the URL is open in the automated test browser it will be automatically surfed by the elements mentioned in the test script.

```
IWebDriver WebDriver;
string[] TotalLinks = new string[100];
TotalLinks[0] = "https://cidco.maharashtra.ete
        TotalLinks[1] = "https://vvcmc.maharas
        TotalLinks[2] = "https://aklmc.maharas
        TotalLinks[3] = "https://umc.maharasht
        TotalLinks[4] = "https://mapo.maharash
        TotalLinks[5] = "https://tribal.mahara
        TotalLinks[6] = "https://udd.maharasht
        TotalLinks[7] = "https://nwcmc.maharas
        TotalLinks[8] = "https://amc.maharasht
```

Figure 5: string array of URL to Scrap.

```
for (int counter = 0; counter <= 8; counter++)
{
    URL = TotalLinks[counter];
    if (URL != "")
    {
                                0 references
        WebDriver.Navigate().GoToUrl(URL);
```

Figure 6: Loop to navigate URL one after another.

A browser will be opened which will show that the browser is being controlled by the automated test software. This means that the browser is being controlled by the test script of the Selenium Automation Framework as shown in the figure 7.

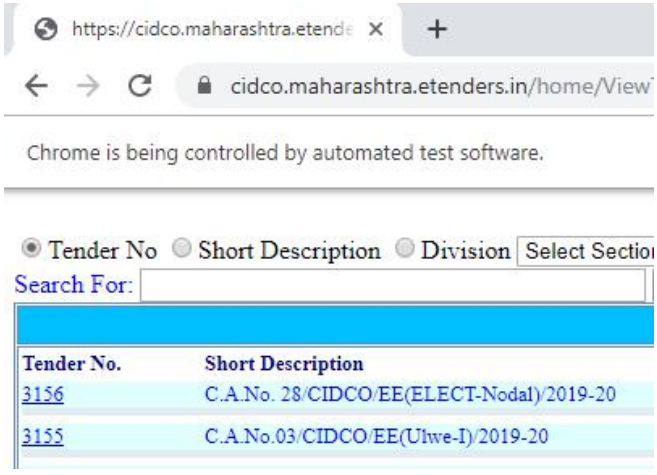

Figure 7: Browser controlled by selenium test script.

B. Select and store scripted document elements in the list by using machine learning techniques.

Here, as per scripted the elements as stored in the list by using machine learning technique Count Vectorizer where the number of occurrences particular element is present in the web page will be counted as shown in figure 8.

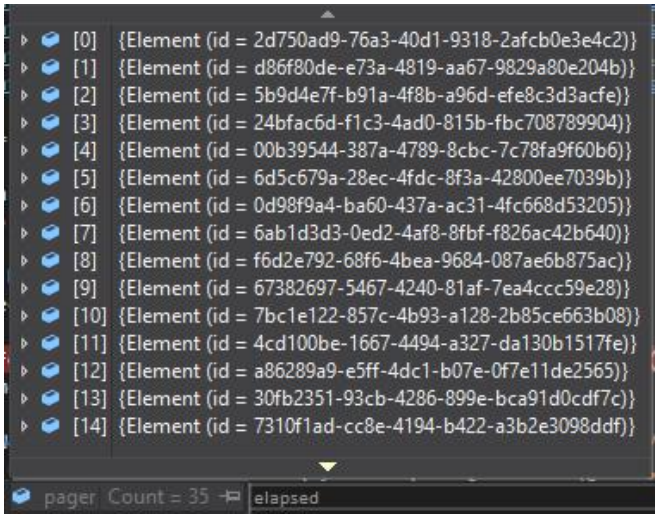

Figure 8: List of elements scripted.

C. Scrap fresh documents by using web client or stream writer.

```
FileData = WebDriver.PageSource;
StreamWriter streamWriter = new StreamWriter
streamWriter.WriteLine(FileData);
streamWriter.Flush();
```

Figure 9: Scrapping using stream writer.

Here in figure 9, file data is just a variable to store the source of the page and the stream writer will write the data and stored it in html format. Here, the documents are stored using one of the Machine Learning techniques which is Forward Selection Wrapper method of Feature Selection where we start the system with having no documents. In each loop of the list, documents are scrapped automatically with their particular unique label name as shown in figure 10 which are fresh and rest will be skipped until the list is empty which will best improve our system.

From the second round of scrapping of the same URL, only the fresh documents will only be scrapped and so that user won't need to surf the whole website daily for only fresh documents and by running this test script user will obtain fresh documents daily until the scripted elements are changed. So, by this it will be less time-consuming task for the user.

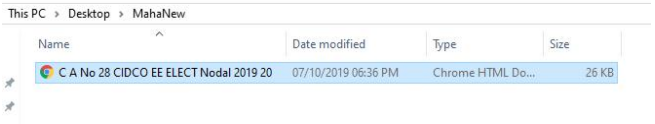

Figure 10: Document scrapped.

V. Conclusion and Future Scope

Selenium Automation Framework is only used for testing purpose for software to run it correctly without errors or bugs. Here the proposed system shows that Selenium Automation can be used not only for testing purpose but it is used for Scrapping purpose also which will be helpful for many users for less time consuming and save manual efforts in browsing data from web pages.

In future, we can create an application by using this and store data from the scrap documents which will be used for marketing purpose. The stored data can also be redirected to provide as services to clients by the companies.

REFERENCES

[1] Nicey Paul and Robin Tommy "An Approach of Automated Testing on Web Based Platform Using Machine Learning and Selenium" International Conference on Inventive Research in Computing Applications (ICIRCA 2018), IEEE 2018.
[2] Shakra Mehak, Rabia Zafar, Sharaz Aslam, Sohail Masood Bhatti "Exploiting Filtering approach with Web Scrapping for Smart Online Shopping" 2019 International Conference on Computing, Mathematics and Engineering Technologies – iCoMET 2019, IEEE 2019.
[3] Miroslav Bures, Martin Filipsky "SmartDriver: Extension of Selenium WebDriver to Create More Efficient Automated Tests", IEEE 2016
[4] Satish Gojarea, Rahul Joshib, Dhanashree Gaigawarec "Analysis and Design of Selenium WebDriver Automation Testing Framework" 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Procedia Computer Science ELSEVIER 2015
[5] Maurizio Leotta, Diego Clerissi, Filippo Ricca, Cristiano Spadaro "Repairing Selenium Test Cases: An Industrial Case Study about Web Page Element Localization" 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation
[6] Prachi Kunte ,Prof. Dashrath Mane "Automation Testing of Web based application with Selenium and HP UFT (QTP)" International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 06 ,June-2017
[7] Md.Foyzur Rahman "BROWSER-BASED AUTOMATION TESTING USING SELENIUM WEBDRIVER" June 2014, TURKU UNIVERSITY OF APPLIED SCIENCES THESIS
[8] Jagannatha S, Niranjanamurthy M, Manushree SP, Chaitra GS "Comparative Study on Automation Testing using Selenium Testing Framework and QTP" International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue. 10, October 2014
[9] Renu Patil, Rohini Temkar "Intelligent Testing Tool: Selenium Web Driver" International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 06 ,June -2017
[10] Niranjanamurthy M, Arun Kumar R, Sahana Srinivas, Manoj RK "Research Study on Web Application Testing using Selenium Testing Framework" International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue. 10, October 2014
[11] Insha Altaf, Jawad Ahmad Dar, Firdous ul Rashid, Mohd. Rafiq "SURVEY ON SELENIUM TOOL IN SOFTWARE TESTING" IEEE 2015
[12] Paruchuri Ramya, Vemuri Sindhura, P Vidya Sagar "Testing using Selenium Web Driver" IEEE 2017
[13] Mr. Dashrath Mane, Gaurav Bhadekar, Santosh Salunkhe "TEXT AND KEYWORD DRIVEN AUTOMATION TESTING USING SELENIUM WEB DRIVER" International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 07, July 2016

[14] Andrea Stocco, Maurizio Leotta, Filippo Ricca, Paolo Tonella "Why Creating Web Page Objects Manually If It Can Be Done Automatically?" IEEE 2015
[15] https://www.webharvy.com/articles/what-is-web-scraping.html.
[16] https://en.wikipedia.org/wiki/Selenium_(software).
[17] https://medium.com/@joshsungasong/natural-language-processing-count-vectorization-and-term-frequency-inverse-document-frequency-49d2156552c1
[18] https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/
[19] https://docs.microsoft.com/en-us/dotnet/api/system.net.webclient?view=netframework-4.8
[20] https://www.guru99.com/c-sharp-stream.html