CEPAT Journal of Computer Engineering: Progress, Application and Technology
Vol. 1, No. 2, August 2022, pp. 16-23
ISSN: 2963-6728, DOI: https://doi.org/10.25124/cepat.v1i02.5226

Application of web scraping on forecast report form of geomagnet

Setyanto Cahyo Pranoto, L M Musafar, Harry Bangkit, Anton Winarko
[1]Research Center for Space, National Research and Innovation Agency, Indonesia

Article Info

Article history:

Received July 31, 2022
Revised August 22, 2022
Accepted August 23, 2022

Keywords:

Web scraping
Data extraction
Forecast Report Form
Swifts
Black box testing

ABSTRACT

The rapid growth of information encourages the development of new technologies that enable data and information processing to be streamlined. One of them is in terms of obtaining information from the website. Most of the current studies targeting this task are mostly about automated web data extraction. Web scraping is a process of extracting valuable and interesting text information from web pages. To realize space weather prediction information services, including geomagnetic activity, the Forecast Report Form of geomagnetic (FRF) application was developed using the web scraping method. Black box testing is used to test the web scrapping process. The result shows that web scraping has been successfully implemented for data retrieval from websites and is reliable for collecting large amounts of data more quickly, automatically, and efficiently with the ability to generate expected results.

Corresponding Author:

Setyanto Cahyo Pranoto
Research Center for Space
National Research and Innovation Agency
Bandung, Indonesia
Email: sety009@brin.go.id

1. INTRODUCTION

In the digital era, the dependence on technology encourages the acceleration of information delivery. This includes the delivery of information related to space weather. This space technology is strongly influenced by space weather, which is sourced from solar activity. Solar activity triggers high-speed charged particles that carry magnetic fields and energy into interplanetary space that potentially reach Earth's orbit. This contributes to geomagnetic disturbances and affects human technology such as satellite discharges and power grid disturbances due to Geomagnetically Induced Current (GIC) [1].

With these risks, it is necessary to have a forecast or prediction information service, about what, when, where, and how space weather disturbances have the potential to cause technological damage and can disrupt human life. The Space Weather Information and Forecast Services (SWIFtS) activity initiated by the LAPAN Space Science Center is a real step in realizing space weather prediction information services, including geomagnetic activities.

In the analysis process, evaluation, and prediction of geomagnetic activity, the Forecast Report Form (FRF) is used by the forecaster team to compile data from various sources, as well as analysis material to predict geomagnetic activities for the next 24 hours. There are two types of data used for this process, namely the processed data and supporting data taken from reference websites such as *wdc.kugi.kyoto-u.ac.jp* and *www.swpc.noaa.gov* which contain information related to geomagnetic index data and space weather data. To be able to use the data automatically, the web scraping method is applied. Web scraping is an activity carried out to retrieve certain data in a semi-structured manner from a website page [2], [3].

This paper describes the application of web scraping to the automatic filling of FRF geomagnetic. A process needs to be built to analyze the document before starting to fetch a large amount of data on a website more quickly.

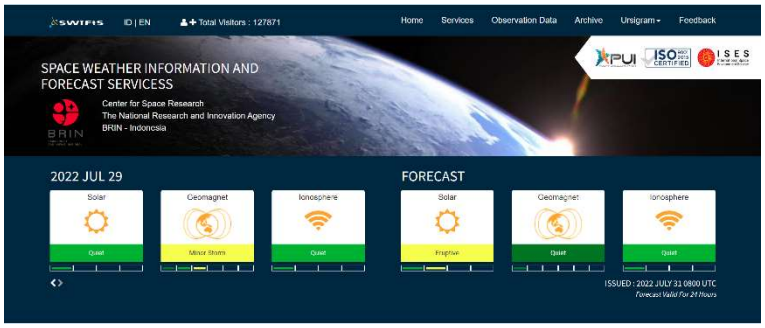


Figure 1. SWIFtS website (Space Weather Information and Forecast Services).

2. METHOD

2.1. Web Scraping

Web scraping, web harvesting, or web data extraction are activities carried out to extract certain semi-structured data from a web page. These pages are generally built using a markup language such as Hyper Text Markup Language (HTML) or Extensible Hyper Text Markup Language (XHTML). A process needs to be established to analyze the document before starting to collect data. Web scraping helps in collecting large amounts of data more quickly and automatically [4], [5]. Web scraping cannot be included in the field of data mining, as data mining indicates an understanding of patterns or trends from the large amount of data obtained. Meanwhile, web scraping only focuses on how to obtain data from a data source with varying data sizes [6].

Generally, scraping techniques are implemented in a system to make processes that should be done manually become automatic. When we encounter a website that limits the Application Programming Interface (API) quota or even does not provide it at all, then browsing the website will be needed as a step to retrieve data and save it in Binary format, Microsoft Excel, JSON, or other formats. With the growing need for web scraping, several automation techniques are commonly used, including:

1. HTML parsing is one of the most widely used techniques in web parsing. HTML parsing is usually done via JavaScript and targets linear and nested HTML pages. This quick method identifies the HTML script of the website, which was possible to be carried out manually beforehand. This script is then used to extract text, links, and data.
2. Document Object Mode (DOM) parsing. The contents, types, and structure of an XML file are defined in the DOM. Scrapers who want to know the internal workings of web pages and extract the scripts that run on them can choose to do web scraping via DOM parsing. Specific nodes are collected using a DOM parser and tools like XPath help with scraping a web page.
3. XPATH. XML Path Language or better known as XPath is a query language that works on XML documents. Since XML documents are usually structured in a tree structure, XPath can be used to navigate the document structure by selecting nodes based on various parameters. XPath can also be used in conjunction with DOM parsing to extract entire pages from a website and display them on other websites.
4. Google Sheets can also be used as a scraping tool which is quite popular by utilizing the IMPORT XML function to scrape data from websites. Apart from that, you can also use this command to see if your website is safe from scraping.

Web scraping can also be done with independent programming techniques using programming languages such as Python, Jason, and so on [7]-[9].

2.2. Research Method

The presentation in this activity is focused on how the application of web scraping is carried out. The system development method used in this activity is a linear sequential model [10]. This model is also known as the classic life cycle which is one of the simplest models on the web that involves the following phases; Analysis, Design, Coding, and Testing. This linear sequential model uses a systematic and sequential software development approach that starts at the system level and progresses at all stages of analysis, design, coding, testing, and maintenance. The stages in this model begin with building all elements of the system and

sorting out which parts will be used as software development materials, taking into account their relationship to hardware, users, and databases.

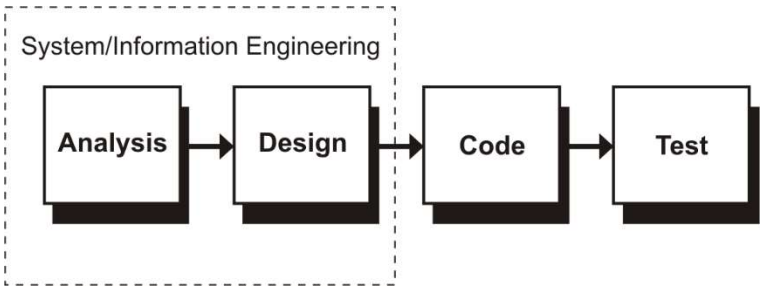


Figure 2. Linear Sequential Model Diagram.

From the many data and information sites related to space weather, in this activity we refer to three websites that are used for data retrieval sites (web scraping), namely;

1. Dst index (*http://wdc.kugi.kyoto-u.ac.jp/dst_realtime/presentmonth/index.html*).
2. AE index (*http://wdc.kugi.kyoto-u.ac.jp/ae_realtime/today/today.html*).
3. Solar wind (*https://www.swpc.noaa.gov/products/ace-real-time-solar-wind*).

The retrieval of data on the website by taking into account the reasons of the site's security system and preventing scraping of the contents and products, so it is necessary to create an algorithm to analyze the content contained on the site [11]. The web scraping applied in this activity only deals with information related to open access link metadata. So that the legal aspects of the application discussed in this article do not violate third-party[12].

3. RESULTS AND DISCUSSION

3.1. System Architecture

Forecast Report Form (FRF) is used by the forecaster team to compile data from various sources, as analysis material to predict geomagnetic activities. To support the effectiveness of these activities, an integrated data management system is needed. A database management system (DBMS) is a collection of interconnected data and a collection of programs to access the data.

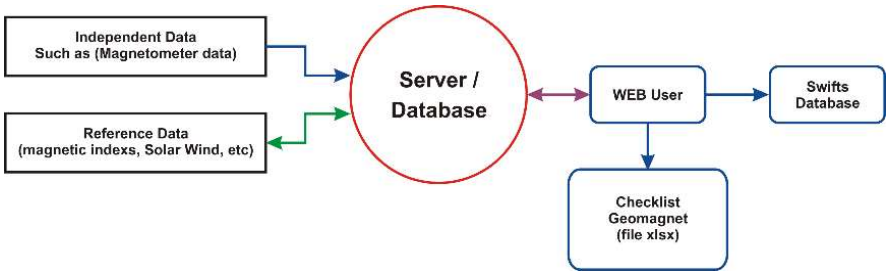


Figure 3. Information system for FRF Geomagnetic.

The diagram flow of the geomagnetic data processing process can be seen in Figure 3. In this process there are two types of data, namely; 1) independent data is indicated by a blue line, and 2) reference data is indicated by a green line. In this independent data processing, all data products are generated from raw data processing and then be corrected before being published as final information. Meanwhile, reference data is data or information obtained from websites by applying the web scraping method. The information taken from the site is an open access link, and the validation of the data is carried out entirely by the party from the site maker. Figure 4 shows index data etc., AE index, and solar wind as reference data.

The independent data and reference data that have been obtained are then stored in a geomagnetic database with output in the form of data such as binary/ASCII, images, and log reports. The main purpose of a database management system is to provide a way to store and retrieve database information easily and efficiently including; efficient access, fast application development, data integrity and security, data uniformity administration, and concurrent access.

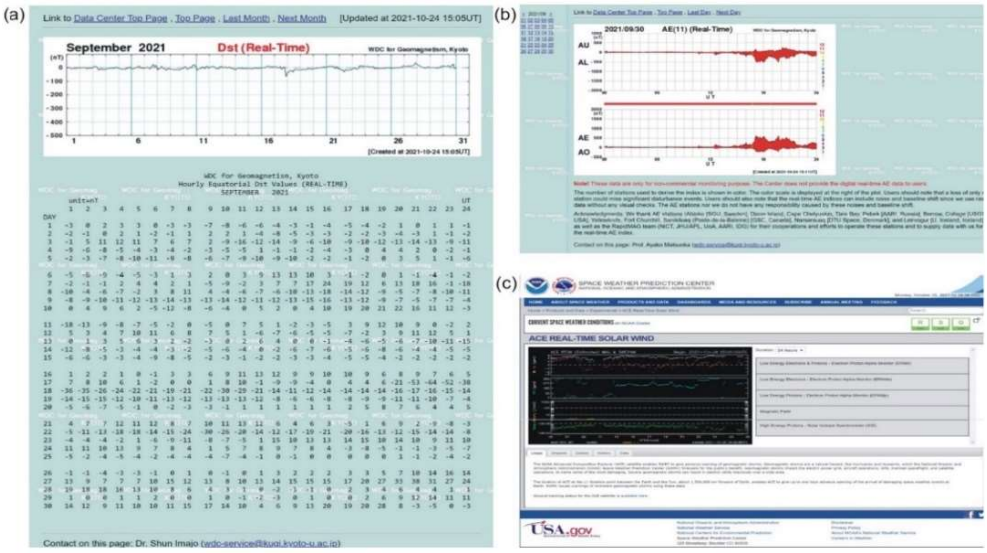


Figure 4. Reference data on website: (a) DST index, (b) AE index, (c) Solar Wind.

3.2. Coding

The scraping technique is implemented in order to make a process that should be done manually become automatic. It only focused on how to obtain data through retrieval and extraction of data with varying data sizes. To be able to do this, coding or programming is needed to translate an algorithm to suit the system design in the application of web scraping. In this activity, the coding of reference data retrieval is made in the form of modules using the python and JSON programming languages.

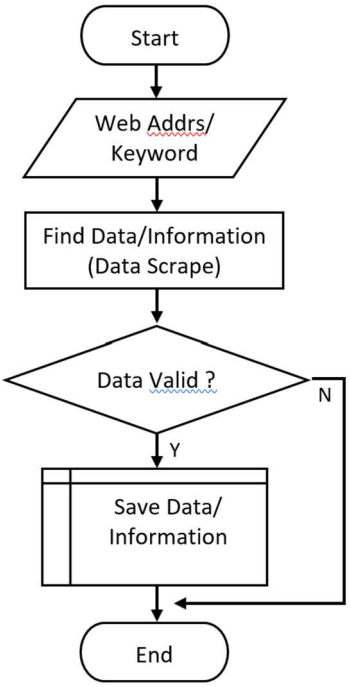


Figure 5. Data search flow on web scraping.

Figure 5 shows the flow of the data retrieval process on web scraping. It starts with entering keywords or the address of the intended site to find the data or information needed. If the sought data is not found, a message will be displayed. The next step is to ensure the validation of the data. Valid data will be stored in a database. The stages of the data search flow are then implemented in the form of coding using the python programming language. Snippets of the coding are shown in Figure 6 and Figure 7.

```
[1]: import os
     import requests
     import datetime
     from PIL import Image

[2]: targetpath = r"d:\Current Work\aeimages"
     if not os.path.isdir(targetpath): os.makedirs(targetpath)

[3]: def get_ae_image(dateobj, baselink, targetpath):
         datestr = datetime.datetime.strftime(dateobj, "%Y%m%d")
         dateymm = datetime.datetime.strftime(dateobj, "%Y%m")
         imagename = "rtae_%s.png" % datestr
         imagelink = baselink + "/" + dateymm  + "/" + imagename
         targetfile = os.path.join(targetpath, imagename)
         try:
             image = requests.get(imagelink)
             fid = open(targetfile,'wb')
             fid.write(image.content)
         except:
             paas
         fid.close()

[4]: dateobj = datetime.datetime.now().date()
     #dateobj = datetime.datetime(2021,2,17).date()
     baselink = "http://wdc.kug1.kyoto-u.ac.jp/ae_realtime"
     get_ae_image(dateobj, baselink, targetpath)
```

Figure 6. A snippet of scraping coding for the AE index.

```
[1]: import requests
     import os
     import datetime
     from bs4 import BeautifulSoup
     import json
     import calendar
     import math

[2]: targetpath = r"d:\Current Work\dstindex"
     if not os.path.isdir(targetpath): os.makedirs(targetpath)

[3]: dateobj = datetime.datetime.utcnow().replace(minute=0, second=0, microsecond=0)
     #dateobj = datetime.datetime(2018,8,1,0,0,0)
     maxday = calendar.monthrange(dateobj.year, dateobj.month)[1]
     print (maxday)
     datestr = datetime.datetime.strftime(dateobj, "%Y%m")
     dstlinkbase = "http://wdc.kug1.kyoto-u.ac.jp/dst_realtime"
     dstlinkname = dstlinkbase + "/" + datestr
     print (dstlinkname)

31
http://wdc.kug1.kyoto-u.ac.jp/dst_realtime/202103

[4]: def split_dst_line(dstline):
         dstday = {}
         tempday = dstline[3::]

         day = int(dstline[0:2])
         partx = tempday[0:32]
         party = tempday[33:65]
         partz = tempday[66:98]
         for xin in range(0,32,4):
             jam = "%02d" % (xin/4 + 1)
             dstval = partx[xin:xin+4]
             dstval = round(float(dstval),0)
             if abs(dstval)>1000: dstval = math.nan
             dstday[jam] = dstval
         for yin in range(0,32,4):
             jam = "%02d" % (yin/4 + 9)
             dstval = party[yin:yin+4]
             dstval = round(float(dstval),0)
             if abs(dstval)>1000: dstval = math.nan
             dstday[jam] = dstval
         for zin in range(0,32,4):
             jam = "%02d" % (zin/4 + 17)
             dstval = partz[zin:zin+4]
             dstval = round(float(dstval),0)
             if abs(dstval)>1000: dstval = math.nan
             dstday[jam] = dstval


         return day, dstday

[5]: dayrange = list(range(1, maxday+1))
     fid = requests.get(dstlinkname)
     soup = BeautifulSoup(fid.text, 'html.parser')
     dataclass = soup.find_all("pre", {"class": "data"})
     dataclass = soup.get_text().split("\n")
     nc = 0
     day = 0
     dstjson = {}
     for dstline in dataclass:
         if dstline and  nc>=30 and len(dstline)>90:
             if int(dstline[0:2]) in dayrange:
                 nowday, dstday = split_dst_line(dstline)
                 nowtimeobj = dateobj.replace(day=nowday)
                 nowtimestr = datetime.datetime.strftime(nowtimeobj,"%Y-%m-%d")
                 dstjson[nowtimestr] = dstday
         nc +=1
         if day>=maxday:
             break
     targetname = "dst_index_%s.json" % datestr
     targetfile = os.path.join(targetpath, targetname)
     print (targetfile)
     fid = open(targetfile,"w")
     fid.write(json.dumps(dstjson, indent=2))
     fid.close()
     #print (json.dumps(dstjson, indent=2))
```

Figure 7. Snippets of scraping code for Dst index.

3.3. Testing of System

The success of a web scraping process can be done by some of test. There are two types of methods that can be used, namely White Box Testing and Black Box Testing [13], [14]. White Box Testing is used to ensure that all commands and conditions on the system are executed at a minimum. White box testing uses two tools, namely a flow graph which is used to describe the flow of the algorithm, and a graph matrix which is used to generate a flow graph. Then the calculation of cyclomatic complexity and graph matrix algorithm. Black Box Testing is the testing of the implementation of web scraping for data retrieval on certain sites carried out on the features in the application by checking whether the application built is running correctly as expected or not. The test is selected based on the problem specification without regard to the internal details of the program and with a top-down approach.

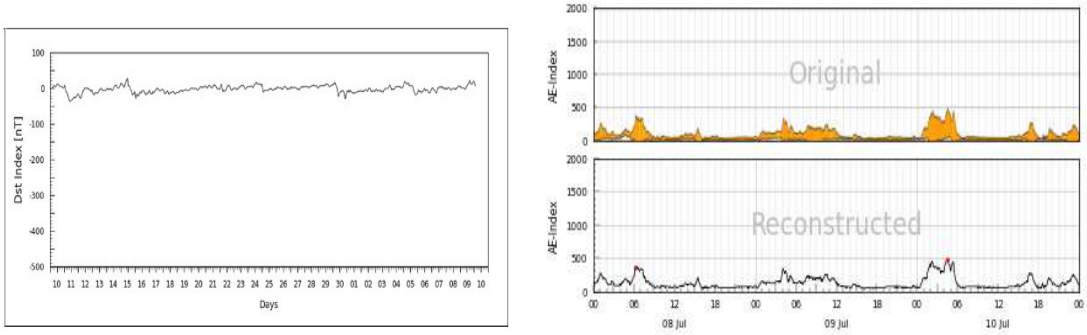
The testing is performed with Black Box testing that focuses on the functional specifications of the software by ignoring the control structure so that its attention is focused on domain information [15]. The scheme of the test refers to the flow of Figure 5, which began with a website search and continued for detailed information from the site. Testing is done by testing the interface part of the information system, each part of the interface is tested to determine whether the system is running according to the expected function. The purpose of this test is to find out errors in the system being made. The following are the test results using the black box technique of evaluation which are displayed in table 1, while Figures 8 - 10 show the output of the test result.

Table 1. Web scraping test results

Testing	Purpose	Status
Primary site search.	Can access the website	Succeed.
Looking for detailed data on the site.	The data generated from the web scraping process is successfully displayed on the detail page.	Succeed.
Scraping numeric data, text, image	The process of retrieving and parsing data from the database went well.	Succeed.

Web scraping involves the creation and implementation of two software programs: a crawler and a scraper. The crawler downloads data from the internet in a systematic manner; the scraper then extracts important information in its raw form from the downloaded data, codes it, and stores it in a database or file according to a user-defined structure and format.

The process of data retrieval (web scraping) is carried out by visiting the website and then searching for an image or data element. The inspection element is done to retrieve important data to be extracted and analyzed as shown in Figure 8. Data for the DST index is taken in the form of text and then reconstructed to obtain data in the form of images, meanwhile the ae index, data is taken in the form of an image which is reconstructed as needed. The solar wind data taken in the web scraping process is an original image without any reconstruction, as shown in Figure 9. The results of the web scraping are then used as analysis material which is stored in the database and displayed in Forecast Report Form as shown in Figure 10. This new file is then evaluated in ways that the initial data presentation on the internet does not allow for. In this process, the metadata or information used is open access so not violate legal aspects.



(a) (b)

Figure 8: Web scraping data for (a) reconstruction of Dst index, (b) reconstruction plot of AE index.

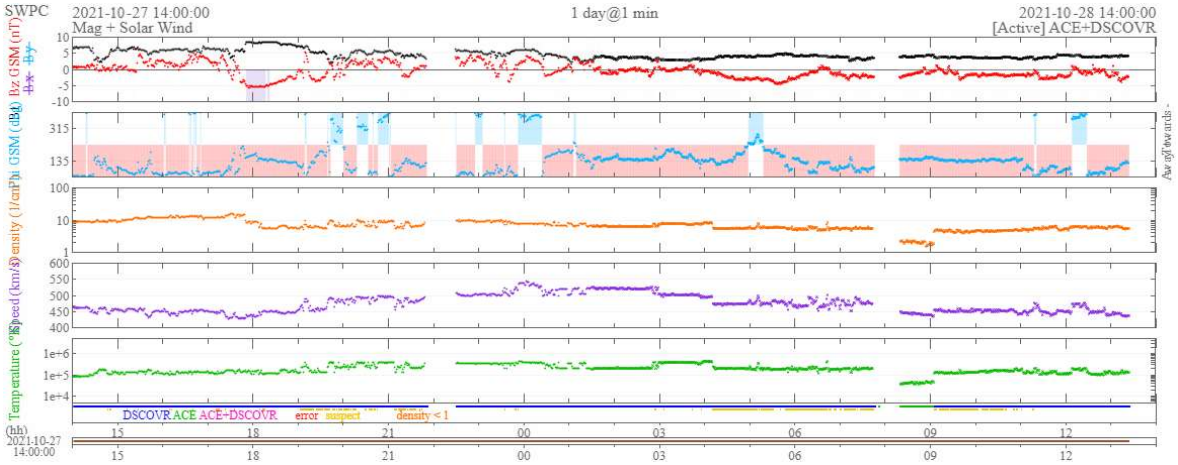


Figure 9. The plot of solar wind data from web scraping.

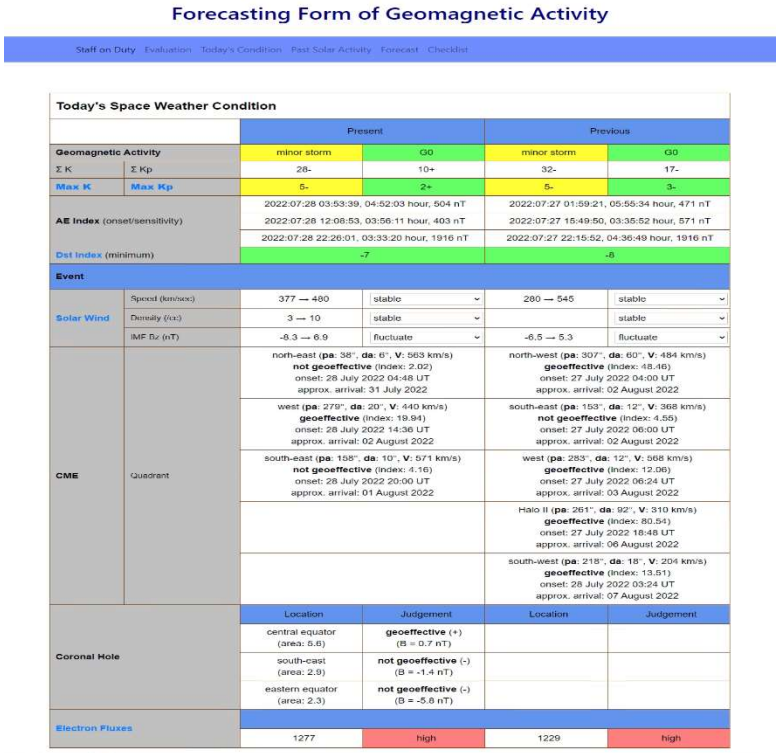


Figure 10. Display of FRF geomagnetic web used web scraping data.

4. CONCLUSION

An application program for filling out the Geomagnet Forecast Report Form (FRF) automatically has been developed for space weather prediction services by applying the web scraping method. The results obtained;

1. Applications developed with web scraping techniques are able to compile, extract and present a number of data and information from websites.
2. Black box testing provides results in accordance with the focused data collection criteria.
3. Information related to metadata is an open access link so that the legal aspects of the application discussed in this article do not violate any parties as long as it is not used for data theft or information manipulation.

ACKNOWLEDGEMENTS

The authors would like to thank the National Oceanic and Atmospheric Administration (NOAA) for providing solar wind parameter data, the World Data Center for Geomagnetism (WDC) Kyoto for Earth's magnetic field data, research group of geomagnetic at the research Center for Space.

REFERENCES

[1] Thomas J. Overbye, "Power Grid Sensitivity Analysis Of Geomagnetically Induced Currents," *IEEE Transactions On Power Systems*, vol. 28, no. 4, 2013.
[2] Moaiad Ahmad Khder, "Web Scraping or Web Crawling: State of Art, Techniques, Approaches, and Application," *Int. J. Advance Soft Compu. Appl,* vol. 13, no. 3, Nov 2021.
[3] Erdinç Uzun, "A Novel Web Scraping Approach Using the Additional Information ObtainedFrom Web Pages," *IEEE Access,* vol.1, no. 1, pp.99, Mar 2020, doi:10.1109/ACCESS.2020.2984503.
[4] T. Karthikeyan and D. Ranjith, "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.
[5] Vidhi Singrodia, Anirban Mitra and Subrata Paul, "A Review on Web Scrapping and its Applications," *International Conference on Computer Communication and Informatics (ICCCI -2019)*, Jan. 2019, pp. 355-360. **DOI:** 10.1109/ICCCI.2019.8821809.
[6] Thomas Bressoud and David White, "Web Scraping," *Introduction to Data Systems*, Springer Nature Switzerland AG, 1st ed, 2020, ch. 22, pp. 681-714.
[7] Ryan Mitchell, "Web Scraping With Python," *Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North*, Sebastopol, CA 95472. ,2018.
[8] David Mathew Thomas and Sandeep Mathur, "Data Analysis by Web Scraping using Python," *Proc. Third International*

Conference on Electronics Communication and Aerospace Technology (ICECA 2019), 2019, pp. 450-454.

[9] Pratiksha Ashiwal, S.R. Tandan, Priyanka Tripathi and Rohit Miri, "Web Information Retrieval Using Python and BeautifulSoup," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, pp. 335-339, 2016.

[10] Pawan Kumar Tanwar and Vishal Goar, "New Approach in Linear Sequential Model for the Development of Software," *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies,* Nov. 2014. article no. 70, pp. 1–4.

[11] Lusiana Citra Dewia, Meilianaa, and Alvin Chandraa, "Social Media Web Scraping using Social Media Developers API and Regex," *Proc. Computer Science 157, 4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI)*, Sep. 2019, pp. 444–449.

[12] V. Krotov and L. Silva, "Legality and ethics of web scraping," Twenty-fourth Americas Conference on Information Systems, New Orleans, 2018.

[13] Alberto Martin-Lopez, Andrea Arcuri, Sergio Segura and Antonio Ruiz-Cort´es, "Black-Box and White-Box Test Case Generation for RESTful APIs: Enemies or Allies?," *32nd International Symposium on Software Reliability Engineering (ISSRE)*, 2021, doi: 10.1109/ISSRE52982.2021.00.

[14] Zahra Abdulkarim Hamza and Mustafa Hammad, "Web and Mobile Applications: Testing using Black and White Box approaches," *Proc. 2nd Smart Cities Symposium (SCS 2019),* Mar. 2019, pp. 200-203. doi:10.1049/cp.2019.0210.

[15] Khamdamov Rustam Khamdamovich and Ibrokhimov Aziz, " Techniques and Methods of Black Box Identifying Vulnerabilities in Web Servers," *International Conference on Information Science and Communications Technologies (ICISCT)*, 2021, doi: 10.1109/ICISCT52966.2021.9670263.