Data Mining applied to Web Robots Detection: A Systematic Mapping

1st Ramon Abilio
Department of Informatics
Federal Institute of São Paulo - IFSP
Boituva, Brazil
ramon.abilio@ifsp.edu.br

2nd Cristiano Garcia
Department of Information Systems
Federal Institute of Santa Catarina - IFSC
Caçador, Brazil
cristiano.garcia@ifsc.edu.br

3rd Victor Fernandes
Department of Informatics
Federal Institute of São Paulo - IFSP
Boituva, Brazil
l.victor@aluno.ifsp.edu.br

***Abstract*—Browsing on Internet is part of the world population's daily routine. The number of web pages is increasing and so is the amount of published content (news, tutorials, images, videos) provided by them. Search engines use web robots to index web contents and to offer better results to their users. However, web robots have also been used for exploiting vulnerabilities in web pages. Thus, monitoring and detecting web robots' accesses is important in order to keep the web server as safe as possible. Data Mining methods have been applied to web server logs (used as data source) in order to detect web robots. Then, the main objective of this work was to observe evidences of definition or use of web robots detection by analyzing web server-side logs using Data Mining methods. Thus, we conducted a systematic Literature mapping, analyzing papers published between 2013 and 2020. In the systematic mapping, we analyzed 34 studies and they allowed us to better understand the area of web robots detection, mapping what is being done, the data used to perform web robots detection, the tools, and algorithms used in the Literature. From those studies, we extracted 33 machine learning algorithms, 64 features, and 13 tools. This study is helpful for researchers to find machine learning algorithms, features, and tools to detect web robots by analyzing web server logs.**

***Index Terms*—Web Usage Mining, Web Server Logs, Machine Learning Algorithms, Feature Extraction, Feature Selection**

I. Introduction

The Internet takes part of our daily life more and more. Humans use the Internet for researching, creating and sharing information through social media, getting in contact with others, having fun and so on. All these activities are normally provided through web pages and web services, which require a service of a web server.

Web servers (e.g. Apache, Nginx) are software responsible for responding requests to sites and web systems available on the Internet. These software listen to a specifics ports in order to perform their responses. In general, these servers record in files each request to their resources (e.g. dynamic web pages, HTML, CSS, images, and fonts).

These records, namely log, contains information on the requests, such as [1]: requestor's IP address, a timestamp, requested resource, amount of transferred bytes, and user-agent. Therefore, these files store several data that can be explored in order to look for access patterns that help understand users' (or web robots') behavior [1].

Web robots started being developed and used in 1993 and, from that time on, the amount of web robots has been increasing and causing problems, such as overload in web servers and waste of bandwidth [2], [3]. Web robots are frequently used by search engines, digital libraries and online marketing to gather information and thus offer the best up-to-date answers to their users [4]. However, they are also used to search for vulnerabilities in web servers, to promote denial of service (DoS) attacks, and to collect sensible data such as personal data and e-mail addresses [5].

Due to the massively requests and the problems caused by them, the detection of web robots' requests is important. This detection can be performed using heuristics or rules, for example, if exists a request to robots.txt file, the requester is a robot. However, modern techniques are based on Web Usage Mining [6] and the use of Data Mining methods to discover patterns on web server logs has been studied over the years.

Therefore, the main objective of this study is to investigate "How have web robots been identified using Data Mining techniques on web logs?". We performed a systematic mapping to observe the use of algorithms, techniques, and tools to detect web robots through log analysis of web servers. In the systematic mapping, we selected 34 studies published between 2013 and 2020. After the data extraction, we found 33 algorithms, 64 features, and 13 tools utilized in those works.

The main contributions of this work are:

- a list of the most used Machine Learning algorithms to detect web robots;
- a list of the most used Features to detect web robots;
- a set of APIs (application programming interface) that can be used to search user-agents or IP to verify if they are robots.

This work is organized as follows. Section II and Section III briefly presents a background and related works. In the Section IV, we describe the systematic mapping in details. The results are detailed and discussed in Section V and the Section VI presents our final remarks and future works.

This work was supported in part by the Voluntary Institutional Program for Scientific and/or Technological Initiation (PIVICT) of the Federal Institute of São Paulo (IFSP)

II. Background

Knowledge Discovery in Databases (KDD) is a systematic methodology to extract implicit useful knowledge from datasets [7]. KDD has 5 major steps [7]: (a) data selection; (b) preprocessing; (c) data transformation; (d) Data Mining; and (e) evaluation. The Data Mining step describes the application of data analysis and algorithms to generate computational models [7]. When the data utilized to feed the process is originated from Web, the Data Mining step is named Web Mining.

Web Mining is the application of Data Mining techniques to discover unknown patterns in web data [6]. According to the data source analyzed, Web Mining can be associated to [6], [8]:

- **Web Content Mining (WCM)**: pattern discovery in the content of web documents;
- **Web Usage Mining (WUM)**: pattern discovery in accesses to web servers;
- **Web Structure Mining (WSM)**: pattern discovery in hyperlinks structure.

The data collected during the use of Web show different behavior patterns, such as: (a) navigation preferences and (b) online customer behavior, which can be used to future improvements in the web site [6]. These information can be recorded in log files in web servers, proxy servers and customers' web browser [1], [6]. Therefore, log files can be analyzed through the Web Usage Mining (WUM) process.

The WUM process comprises [1]: (a) data gathering; (b) data integration; (c) preprocessing; (d) pattern discovery; and (e) patterns analysis. In the preprocessing step, the data cleansing is performed, by removing, for example: accesses to multimedia files, to cascading style sheet (CSS) files, and requests done by web robots [1], [9]. In the pattern discovery step, several techniques can be used, such as association rules mining and clustering. In the pattern analysis step, data visualization techniques and online analytical processing (OLAP) can be used [1].

Although the practice of removing web requests done by web robots in the preprocessing step, some papers highlight the need for improvement of web robots detection [1], [9].

Logs may have different formats depending on the web server configuration. For example, Apache HTTP Server has, by default, a log file for errors and other file for visitors' access (normally named access.log) [10]. When configured with "Combined Log Format", logs in access.log have these fields [10]: client IP address, identity of the client, user identification in case of authenticated access, date/time that the request was received by the web server, method and requested resource, status code sent to the client, number of transferred bytes, page that links to or includes the requested resource (referrer), and user-agent information.

The Apache HTTP Server log files' documentation shows this example of a log registered in access.log file using "Combined Log Format" [10]:

```
127.0.0.1 - frank
[10/Oct/2000:13:55:36 -0700]
"GET /apache_pb.gif HTTP/1.0" 200 2326
"http://www.example.com/start.html"
"Mozilla/4.08 [en] (Win98; I ;Nav)"
```

Logs are registered one per line in the access.log file (we inserted line breaks in the example above). Following, we described each field of the example:

- **127.0.0.1**: Client IP;
- **-**: Identity of the client (in this case, unavailable ("-"));
- **frank**: User identification on authenticated access (when unavailable ("-"));
- **[10/Oct/2000:13:55:36 -0700]**: Date and time that the request was received (Date = 10/Oct/2000 and Time = 13:55:36 (Zone = -0700));
- **"GET /apache_pb.gif HTTP/1.0"**: Requested resource = /apache_pb.gif, Method = GET, Protocol/Version = HTTP/1.0;
- **200**: Status code sent to the client (request resulted in a successful response (codes beginning in 2), a redirection (codes beginning in 3), an error caused by the client (codes beginning in 4), or an error in the server (codes beginning in 5));
- **2326**: Number of bytes transferred to the client;
- **"http://www.example.com/start.html"**: Page that links or includes the requested resource (referrer);
- **"Mozilla/4.08 [en] (Win98; I ;Nav)"**: User-agent information.

Depending on the web page's popularity, an access.log file may have hundreds of thousands of logs.

III. Related Work

An access.log file stores lots of information that can be explored to know visitors' behavior. To explore those data, several WUM methods have been proposed over the years and gained researchers attention.

In 2014, a study [11] presented techniques applied in preprocessing step of Web usage mining with their advantage and disadvantage. In 2017, 2 studies [12] [1] provided overviews of Web Usage Mining (WUM) and explained the process involved in WUM, its applications and tools.

A study published in 2018 describes Web Mining, its different types, tools, and techniques and shows a table with data mining algorithms used in WUM tasks [13]. In 2019, a review of WUM techniques applied in data preprocessing of Web server log with emphasis on data cleaning, user identification, and session identification was published [9].

We can observe that all aforementioned studies focused on WUM process, techniques, and tools. Only in 2020, we found a study dedicated to present an overview of web robots detection [14]. This study showed different approaches and challenges of the three themes of web robot detection techniques: machine learning, honeypots, and online robot detection.

Our work is in the same topic of the others (WUM), but it differs from the others. While they present wide data regarding WUM, we focused on machine learning, features and tools that can be used in preprocessing, pattern discovery and analysis phases of WUM to detect web robots by analysis of web server logs. We also present the common steps followed in the data preparation (session reconstruction, feature extraction and feature selection).

IV. Systematic Mapping

According to [15], a systematic mapping is "a methodology used in order to construct an organized scheme of a field of interest". This methodology consists of steps such as: (a) definition of research question; (b) research conduction; (c) screening of papers; (d) keywording using abstracts; and (e) systematic map.

As the first step, we developed a protocol to the systematic mapping and main sections are described as follows.

A. Definition of Research Question

Web robots make requests to web sites aiming at indexing or seeking security vulnerabilities. All requests are logged by the web server in the same file as the other requests. Ethical web robots requests the robots.txt file at first and identify itself as a bot in the user-agent field. However, malicious web robots do not request robots.txt and do not identify itself. Moreover, they try to emulate human navigation patterns or they inform, in the user-agent field, that they are recognized web robots, such as Google or Bing robots. Therefore, the main problem is to identify web robots requests among human requests since all requests are registered in same log file (access.log).

Aiming at finding studies on web robots identification, we defined this Research Question: How have web robots been identified using Data Mining techniques on web logs?

B. Research conduction: Sources and Search String

We used Google Scholar to obtain studies related to the problem and question applying a search string and limiting the period between 2013 and 2020. The search string used was:

```
(robot OR bot OR scrapy OR crawler
 OR spider OR ''web robots''
 OR indexer OR ''web wanderers'')
AND
(''web server log'' OR ''web logs'')
AND
(detection OR identification
 OR classification)
```

Google Scholar was chosen as the source since it indexes the main digital libraries (e.g. IEEExplore, Springer and so on) and it also makes accessible works that are not indexed on these aforementioned digital libraries. We believe that this makes our search more complete and democratic.

C. Screening of papers for inclusion and exclusion: studies selection

We selected all studies that passed in the criteria of inclusion and exclusion. The inclusion criteria were: (i) The study was published after 2012; (ii) The study was published in Journals or Conference Proceedings; (iii) The study must be accessible in an electronic way; and (iv) The study presents the definition or use of data mining techniques to analyze web server logs and detection web robots requests. The exclusion criteria were: (i) The study is incomplete or has restricted access; (ii) The study did not pass in the inclusion criteria; (iii) The study is not written in English; and (iv) The study is not an article.

We imported the Google Scholar search results in JabRef Software[1] and performed the studies selection in two steps. In the first step, the researchers executed the selection process reading the title and abstract of each reference and checking the criteria of inclusion and exclusion. The researcher classified the reference in one of this groups: Included; Excluded; Non-article; Duplicated; or Non-English. In the second step, the researchers read all the included references and extracted the data related to the Question.

D. Keywording using abstracts

Yet according to [15], keywording (i.e. collecting keywords on the paper) using abstract is necessary to get a proper understanding on the nature and contributions of a work. In our case, we adapted this strategy, taking a deeper look (not only at the abstract, but also at the methodology and conclusion sections) to extract information about the features, the machine learning algorithms and the tool used in each paper. The results of the systematic mapping can be verified in the next Section.

V. Results and discussion

In this section, we present and discuss the systematic mapping results. First of all, we present the results of the papers selection. After that, extracted machine learning algorithms and features used to identify human and robots session are shown. At last, we show the process and tools used by the selected papers on sessions classification/clustering.

A. Selected Papers

We performed the search in September/2020 and obtained 2150 references. In the first step of the selection, we included 60 references and excluded 2090, that were mostly non-articles (e.g. books, patents or MSc/PhD thesis). In the second step, we included 34 studies and excluded 26 (Table I) that did not pass by the inclusion criteria.

Fig. 1 presents the number of publications per publication venue. We can notice that 17 selected references were published in Conference Proceedings and 17 were published in Journals. With the exception of "Journal of Network and Computer Applications", with 2 references published, all the other Conference Proceedings and Journals published 1 selected reference.

[1] https://www.jabref.org/

TABLE I
NUMBER OF SELECTED OR EXCLUDED REFERENCES BY YEAR

Year	Status	
	Included	*Excluded*
2013	7	3
2014	4	2
2015	5	3
2016	2	3
2017	4	4
2018	6	6
2019	2	5
2020	4	0
Total	**34**	**26**

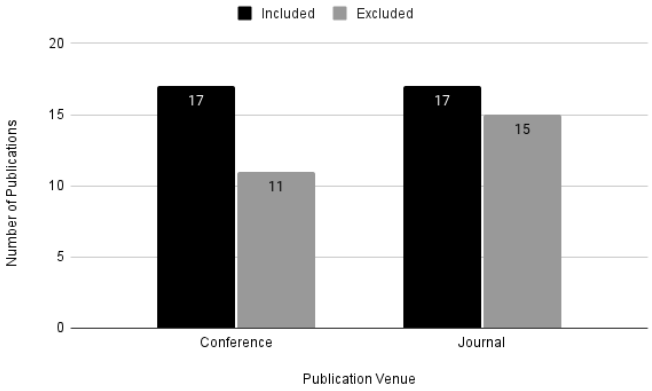

Fig. 1. Number of Publications per Publication Venue

TABLE II
NUMBER OF INCLUDED OR EXCLUDED PAPERS PER PUBLISHER

Publisher	Status	
	Included	*Excluded*
ieeexplore.ieee.org	11	6
link.springer.com	4	4
www.sciencedirect.com	4	1
dl.acm.org	3	1
www.academia.edu	2	1
academicscience.co.in	1	0
bib.irb.hr	1	0
onlinelibrary.wiley.com	1	0
search.proquest.com	1	0
www.ijarcs.info	1	0
www.ijecs.in	1	0
www.informatica.si	1	0
www.isecure-journal.com	1	0
www.jocm.us	1	0
www.scirp.org	1	0
ist.psu.edu	0	2
papers.ssrn.com	0	1
www.arxiv.org	0	1
www.csjournals.com	0	1
www.igi-global.com	0	1
www.ijarcsse.com	0	1
www.ijcaonline.org	0	1
www.ijics.com	0	1
www.researchgate.net	0	1
www.scs-europe.net	0	1
www.thesai.org	0	1
www.tmrfindia.org	0	1
Total	**34**	**26**

Table II shows the number of papers per publisher. For example, we included 11 papers published by IEEEXplore (ieeexplore.ieee.org) and excluded 6 papers. We can note that Google Scholar indexes lots of publishers, since we found references from 27 different sources. From those sources, we include 34 papers published by 15 different publishers.

The main sources regarding the number of included papers were (Table II): IEEEXplore (ieeexplore.ieee.org), Springer (link.springer.com), Science Direct (www.sciencedirect.com), and ACM Digital Library (dl.acm.org). In this work, we have publishers without selected papers, but we opted to show all publishers due to their relevance to the topic.

Table III shows reference, title, and year of publication of the 34 included papers for full read and data extraction.

B. Machine Learning Algorithms

We found 33 machine learning algorithms in the extracted data. We noticed that authors used supervised and unsupervised algorithms for the performed tasks. The 10 most frequent machine learning algorithms in the studies are listed in the Table IV.

The most cited algorithms were SVM (cited in 9 papers) and Decision Trees (cited in 8 papers). Several works tested different classifiers and compared their results regarding the sessions classification in robots or human [20], [27], [34], [39], [45] or even used an Ensemble [30], [44]. We also found papers that evaluated unsupervised algorithms regarding their performance on grouping sessions [40]–[42].

We observed a change (or an evolution) of techniques applied to identification of human and robots sessions. We found 8 works between 2013 and 2017 that used heuristics based, for example, on the access to robots.txt file, user-agent field analysis (presence of words: robot, bot, crawler, spider...), referrer field blank (or with "-"), and the number (or percentage) of requests with error codes to identify sessions of robots [16], [17], [19], [21], [23], [24], [29], [36]. After this period, heuristics were used to assign ground-truth labels to sessions and those sessions were classified by machine learning algorithms.

To increment the heuristics and get better results, in some papers, user-agents and client's IPs were searched on APIs to verify if they are robots. Following, we have, respectively, 5 APIs found in extracted data and other 5 sources added ad hoc by the authors of this study:

- http://www.user-agents.org/index.shtml
- http://www.robotstxt.org/db.html
- https://botsvsbrowsers.org/
- https://www.iplists.com
- http://www.useragentstring.com/index.php
- https://udger.com/resources/online-parser
- https://developers.whatismybrowser.com/
- https://useragents.io/parse
- https://user-agents.net
- https://anti-hacker-alliance.com/

TABLE III
REFERENCE, TITLE, AND YEAR OF PUBLICATION OF SELECTED PAPERS

Ref.	Title	Year
[16]	Identification and characterization of crawlers through analysis of web logs	2013
[17]	Access patterns for robots and humans in web archives	2013
[18]	Detecting Impolite Crawler by Using Time Series Analysis	2013
[19]	A comparison of web robot and human requests	2013
[20]	Detecting anomalous Web server usage through mining access logs	2013
[21]	Mining web logs to identify search engine behaviour at websites	2013
[22]	An integrated approach to defence against degrading application-layer DDoS attacks	2013
[23]	Detection and confirmation of web robot requests for cleaning the voluminous web log data	2014
[24]	Analysis of Aggregated Bot and Human Traffic on E-Commerce Site	2014
[25]	A Supplementary Method for Malicious Detection Based on HTTP-Activity Similarity Features	2014
[26]	A density based clustering approach to distinguish between web robot and human requests to a web server	2014
[27]	Lino - An Intelligent System for Detecting Malicious Web-Robots	2015
[28]	Optimized Distributed Association Mining (ODAM) Algorithm for detecting Web Robots	2015
[29]	A Comparative Analysis of Browsing Behavior of Human Visitors and Automatic Software Agents	2015
[30]	Agglomerative approach for identification and elimination of web robots from web server logs to extract knowledge about actual visitors	2015
[31]	HTTP-sCAN: Detecting HTTP-flooding attack by modeling multi-features of web browsing behavior from noisy web-logs	2015
[32]	An integrated method for real time and offline web robot detection	2016
[33]	HTTP Flooding Attack Detection by Modeling Features of Web Browsing behavior from Web Log	2016
[34]	Website Navigation Behavior Analysis for Bot Detection	2017
[35]	A study of different web-crawler behaviour	2017
[36]	Analysis of Robot Detection Approaches for Ethical and Unethical Robots on Web Server Log	2017
[37]	A soft computing approach for benign and malicious web robot detection	2017
[38]	Bot or Not? A Case Study on Bot Recognition from Web Session Logs	2018
[39]	User behavior analytics-based classification of application layer HTTP-GET flood attacks	2018
[40]	Performance Evaluation of Large Data Clustering Techniques on Web Robot Session Data	2018
[41]	Categorization Performance of Unsupervised Learning Techniques for Web Robots Sessions	2018
[42]	Performance Evaluation of Density-Based Clustering Methods for Categorizing Web Robot Sessions	2018
[43]	A System Framework for Efficiently Recognizing Web Crawlers	2018
[44]	Towards a framework for detecting advanced Web bots	2019
[45]	A Hybrid Approach for Recognizing Web Crawlers	2019
[14]	An Overview of Web Robots Detection Techniques	2020
[46]	Determination of User Navigational Patterns from Server Log Files using Hadoop Techniques	2020
[47]	Bot recognition in a web store: an approach based on unsupervised learning	2020
[48]	Identifying legitimate Web users and bots with different traffic profiles–an Information Bottleneck approach	2020

TABLE IV
MACHINE LEARNING ALGORITHMS EXTRACTED FROM THE SELECTED WORKS AND THEIR RESPECTIVE FREQUENCIES (10 MOST FREQUENT)

Algorithm/Technique	Frequency
Support Vector Machines (SVM)	9
Decision Trees (C4.5, J48)	8
Naïve Bayes (NB)	5
Neural Networks (NN)	5
Random Forests (RF)	5
DBSCAN	3
K-Means	3
K-Nearest Neighbor Classifier (K-NN)	3
Logistic Regression	3
Multi-layer Perceptron Neural Network	3

TABLE V
FEATURES EXTRACTED FROM THE SELECTED WORKS AND THEIR RESPECTIVE FREQUENCIES (10 MOST FREQUENT)

Feature	Frequency
number of requests in the session (numerical)	17
% of requests with empty referrer (numerical)	17
% of requests with status codes 4xx (numerical)	16
% of requests using HEAD method (numerical)	14
robots.txt or other trap file access (boolean)	13
session duration in seconds (numerical)	12
total of transferred bytes (also known as volume or bandwidth)	10
% of requests to image files	9
Image to Page Ratio (numerical)	9
% of requests with status codes 5xx (numerical)	9

C. Features

Another relevant information obtained from the data extraction was that each work performed its tasks using different sets of features. We extracted 64 features and the 10 most frequent features are listed in the Table V.

During the data extraction, we grouped features with similar meaning. For example, we grouped *volume*, *total volume*, and *bandwidth* as *total of transferred bytes*. Therefore, the nominal number of features found is higher than 64. The number of used features ranges from 2 to 50 per work.

A study published in 2015 [27] developed a system to collect data from the client-side and server-side. The authors used, for example, hidden links and a web form in that system to identify web robots with other 6 features such as: number of requests in the session, a boolean feature that indicates if all the requests have empty referrer, and a boolean feature that indicates changes from one session to another subsequent session. After the feature selection, 5 features were selected: a boolean feature that indicates if the client has filled or not the

fake form; the feature that indicates a change from one session to another; session duration in seconds; a boolean feature that indicates if the robots.txt file was requested; and a feature with a percentage of access to hidden.

In a study published in 2017 [37], 2 datasets and a total of 30 features were used in the experiment. The applied strategy selects the features according to the dataset. In the experiments, 11 features were selected for dataset 1 and 9 features were selected for dataset 2. This feature selection strategy allowed better results. Examples of the selected features per dataset are: (i) Dataset 1: % of requests demanded between 12 a.m. and 7 a.m.; % of requests with empty referrer; and trapFileRequest (it shows that if a trap file was requested (robots.txt, sitemap.xml, ...)); and (ii) Dataset 2: page popularity index; % of requests with empty referrer; and % of requests performed between 12 a.m. and 7 a.m.

The authors of a study [44] with 23 features used 3 methods of feature selection. They performed an experiment with 4 classifiers and an ensemble and they divided the dataset in 2 subsets. The feature selection was based on the classifier and on the subset. Thus, they obtained different sets of features since each feature might contribute differently in the performance of each classifier. As an example of features that appears simultaneously in the subsets of features of each classifier, we have: % of requests with empty referrer, number of HTTP POST requests, % of consecutive sequential HTTP requests, max requests per page to the same page in a session, image to page ratio, and search engine referrer (binary feature that checks if a session has at least one request with a known search engine referrer).

Another study [48] initiated with 50 features and a dataset of a real e-commerce. After applying feature selection with Fischer Score, the authors selected 6 features to proceed the experiments. Those 6 features were: % of requests with empty referrer, % of pages with empty referrer, % of page requests, % of image requests, embedded objects to page ratio, and maximum number of embedded requests per page.

We can note that different features were selected in each study probably due to used machine learning algorithms, datasets, and features. We also observed that feature selection was performed based on the datasets [37], on the algorithms [44], or even irrespectively of the learning algorithm [27], [48].

D. Process and Tools

In general, the analyzed papers used the following steps in their process of human and robots identification:

- Logs gathering;
- Data extraction from the requests;
- Session (re)construction;
- Feature extraction;
- Feature selection;
- Classification or Clustering;
- Results Evaluation.

As feature selection methods, we observed the use of: Information Gain, Gain Ratio, Symmetrical Uncertainty, and Relief Method [27]; Forward Approximation Algorithm [37]; Principal Component Analysis (PCA), chi-square, and Sequential Feature Selection [44]; and Fischer Score [48].

From the 34 selected papers, 27 indicated the used tools. In total, 13 tools were mentioned by the authors. We listed all tools with the number of citations (frequency) in the Table VI.

TABLE VI
TOOLS EXTRACTED FROM THE SELECTED WORKS AND THEIR RESPECTIVE FREQUENCIES

Tool	Frequency
Weka	8
A tool developed by the authors (C++, Java)	3
Elki	2
JBIRCH	2
Matlab	2
Python script	2
R	2
AWStats	1
Hadoop	1
Microsoft Azure Cloud-based Machine Learning Framework	1
Scala script	1
Python script + Scikit Learn	1
Shell script	1

We can observe that Weka was the most used tool with 8 citations. Weka is an open source software with tools for data preparation and a collection of machine learning algorithms and it makes sense to appear this frequently because of its ease to use and to perform experiments. It can also be integrated with Java applications.

VI. CONCLUSIONS

Web robots are in use since 1993 for the most diverse objectives. We have robots specialized in images, files, and content indexing. Besides, we have malicious or non-ethical robots used, for example, to scan web vulnerabilities.

Several works have been published on the identification of human and robots web site/systems access. Initially, the main goal was to identify human access and to understand the access behavior (e.g. steps performed since a search of product in a web store to the purchase). For this purpose, the robots requests were removed from the dataset. However, with the growth of the number of robots, the focus changed to identification and classification of the robots.

Therefore, the main objective of this work was the identification of machine learning methods used in the analysis of web server logs to detect web robots access. We found 33 machine learning algorithms used with more than 64 features and we recognized a general process and tools for the web server logs data mining.

The analyzed papers used machine learning models in offline settings, i.e. the data were collected and processed/analyzed out of the web server while the access by the visitors keep going on. This means that data will be collected during a specific period of time and the machine learning model will be learning the specific patterns from that data. If the pattern change (and it probably will), the model will have to be re-trained with more recent data. This change in

the data patterns (or data distributions) is called concept drift [49]. Therefore, to handle this situation, there are incremental approaches. An incremental model (i.e. model that learns on-the-fly, without retraining [50], [51]) would be the ideal for this scenario and that sort of model could be more deeply studied.

We observed that each method and set of features were selected based on the analyzed logs and had different performances. Thus, it is difficult to compare the results among the papers. We also noticed that the session labeling is a challenge, since we have a huge amount of data that is almost impossible to label manually and the use of heuristics and APIs may not detect all robots due to their behavior. The correct labeling is crucial to the machine learning algorithms.

From the analyzed data, we can conclude that the research in the topic can evolve, since the studies were not conclusive on which machine learning methods or features can be effectively used and there exists a gap on online classification models studies.

As other systematic mapping studies, this work has limitations, for example: the time elapsed from the date when the search was performed and the date of results publication; and the tool used in the search (Google Scholar) may not index relevant papers. Thus, important relevant papers might have been published and could not be covered by our search.

As future work, we suggest: the study of incremental classification methods [50], [51]; the definition of sets of features that could be selected according to web server logs; and the development of a (visual) tool to help the analysis of web robots requests.

REFERENCES

[1] N. Kandpal, R. Sinha, and M. Shekhawat, "A survey on web usage mining: Process, application and tools," *Suresh Gyan Vihar University Journal of Engineering & Technology*, vol. 3, no. 1, pp. 19–25, 2017.

[2] C. Bomhardt, W. Gaul, and L. Schmidt-Thieme, "Web robot detection - preprocessing web logfiles for robot detection," in *New Developments in Classification and Data Analysis*, H. Bock and et al, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 113–124.

[3] D. Doran, "Detection, classification, and workload analysis of web robots," Ph.D. dissertation, University of Connecticut, Storrs, CT, 2014, accessed: 2021-06-16. [Online]. Available: https://opencommons.uconn.edu/dissertations/348

[4] F. Menczer, G. Pant, P. Srinivasan, and M. E. Ruiz, "Evaluating topic-driven web crawlers," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 241–249.

[5] Y. Sun, I. Councill, and C. Giles, "The ethicality of web crawlers," in *Proceedings of International Conference on Web Intelligence and Intelligent Agent Technology (IEEE/WIC/ACM)*, 2010, pp. 668–675.

[6] G. Chang, M. Healey, J. McHugh, and J. Wang, *Mining the World Wide Web. The Information Retrieval Series*. Boston: Springer, 2001.

[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[8] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan, "Web usage mining: discovery and applications of usage patterns from web data," *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 12–23, 2000.

[9] M. Srivastava, A. Srivastava, and R. Garg, "Data preprocessing techniques in web usage mining: A literature review," in *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, 2019.

[10] Apache Software Foundation. Apache HTTP server version 2.4 - log files. Accessed: 2021-06-23. [Online]. Available: https://httpd.apache.org/docs/2.4/logs.html

[11] M. Srivastava, R. Garg, and P. Mishra, "Preprocessing techniques in web usage mining: A survey," *International Journal of Computer Applications*, vol. 97, no. 18, pp. 1–9, 2014.

[12] R. Rao and J. Arora, "A survey on methods used in web usage mining," *International Research Journal of Engineering and Technology IRJET*, vol. 4, no. 5, pp. 2627–2631, 2017.

[13] M. Mughal, "Data mining: Web data mining techniques, tools and algorithms: An overview," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, pp. 208–2015, 2018.

[14] H. Chen, H. He, and A. Starr, "An overview of web robots detection techniques," in *Proceedings of the International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2020)*, jun 2020, pp. 1–6.

[15] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*, 2008, pp. 1–10.

[16] N. Algiriyage, S. Jayasena, G. Dias, A. Perera, and K. Dayananda, "Identification and characterization of crawlers through analysis of web logs," in *Proceedings of the 8th International Conference on Industrial and Information Systems*, dec 2013, pp. 150–155.

[17] Y. A. AlNoamany, M. C. Weigle, and M. L. Nelson, "Access patterns for robots and humans in web archives," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries - JCDL '13*, 2013.

[18] Z. Chen and W. Feng, "Detecting impolite crawler by using time series analysis," in *Proceedings of the 25th International Conference on Tools with Artificial Intelligence*, nov 2013.

[19] D. Doran, K. Morillo, and S. S. Gokhale, "A comparison of web robot and human requests," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, aug 2013.

[20] T. Gržinic, T. Kišasondi, and J. Šaban, "Detecting anomalous web server usage through mining access logs," *Central European Conference on Information and Intelligent Systems*, 2013.

[21] J. Jose and P. S. Lal, "Mining web logs to identify search engine behaviour at websites," *Informatica*, vol. 37, no. 2013, pp. 381–386, 2013.

[22] D. Stevanovic and N. Vlajic, "An integrated approach to defence against degrading application-layer ddos attacks," in *Proceedings of the 12th International Conference on Security and Management*, 2013.

[23] T. H. Sardar and Z. Ansari, "Detection and confirmation of web robot requests for cleaning the voluminous web log data," in *Proceedings of the International Conference on the IMpact of E-Technology on US (IMPETUS 2014)*, jan 2014.

[24] G. Suchacka, "Analysis of aggregated bot and human traffic on e-commerce site," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, sep 2014.

[25] M. Tran and Y. Nakamura, "A supplementary method for malicious detection based on http-activity similarity features," *Journal of Communications*, vol. 9, no. 12, 2014.

[26] M. Zabihi, M. V. Jahan, and J. Hamidzadeh, "A density based clustering approach to distinguish between web robot and human requests to a web server," *The ISC International Journal of Information Security (ISeCure)*, vol. 6, no. 1, pp. 1–13, 2014.

[27] T. Gržinić, L. Mršić, and J. Šaban, "Lino - an intelligent system for detecting malicious web-robots," in *Asian Conference on Intelligent Information and Database Systems*. Springer International Publishing, 2015, pp. 559–568.

[28] A. D. Jagtap and V. Kadroli, "Optimized distributed association mining (odam) algorithm for detecting web robots," *International Journal of Engineering and Computer Science (IJECS)*, vol. 4, no. 7, pp. 13 196–13 200, 2015.

[29] D. Sisodia, S. Verma, and O. Vyas, "A comparative analysis of browsing behavior ofhuman visitors and automatic software agents," *American Journal of Systems and Software*, 2015.

[30] D. S. Sisodia, S. Verma, and O. P. Vyas, "Agglomerative approach for identification and elimination of web robots from web server logs to extract knowledge about actual visitors," *Journal of Data Analysis and Information Processing*, vol. 03, no. 01, pp. 1–10, 2015.

[31] J. Wang, M. Zhang, X. Yang, K. Long, and J. Xu, "HTTP-sCAN: Detecting HTTP-flooding attack by modeling multi-features of web browsing

behavior from noisy web-logs," *China Communications*, vol. 12, no. 2, pp. 118–128, feb 2015.

[32] D. Doran and S. S. Gokhale, "An integrated method for real time and offline web robot detection," *Expert Systems*, vol. 33, no. 6, pp. 592–606, nov 2016.

[33] A. Verma and D. Xaxa, "Http flooding attack detection by modeling features of web browsing behavior from web log," *International Journal of Innovations & Advancement in Computer Science (IJIACS)*, vol. 5, no. 6, pp. 154–159, 2016.

[34] R. Haidar and S. Elbassuoni, "Website navigation behavior analysis for bot detection," in *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA 2017)*, oct 2017, pp. 60–68.

[35] A. Menshchikov, A. Komarova, Y. Gatchin, A. Korobeynikov, and N. Tishukova, "A study of different web-crawler behaviour," in *Proceedings of the 20th Conference of Open Innovations Association (FRUCT)*, apr 2017, pp. 268–274.

[36] M. Srivastava, A. kumar Srivastava, R. Garg, and P. K. Mishra, "Analysis of robot detection approaches for ethical and unethical robots on web server log," *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 8, no. 5, pp. 1132–1134, May 2017.

[37] M. Zabihimayvan, R. Sadeghi, H. N. Rude, and D. Doran, "A soft computing approach for benign and malicious web robot detection," *Expert Systems with Applications*, vol. 87, pp. 129–140, nov 2017.

[38] S. Rovetta, A. Cabri, F. Masulli, and G. Suchacka, "Bot or not? a case study on bot recognition from web session logs," *Quantifying and Processing Biomedical and Behavioral Signals*, pp. 197–206, aug 2018.

[39] K. Singh, P. Singh, and K. Kumar, "User behavior analytics-based classification of application layer HTTP-GET flood attacks," *Journal of Network and Computer Applications*, vol. 112, pp. 97–114, jun 2018.

[40] D. S. Sisodia, R. Borkar, and H. Shrawgi, "Performance evaluation of large data clustering techniques on web robot session data," in *Machine Intelligence and Signal Analysis*. Springer, aug 2018, vol. 748, pp. 545–553.

[41] D. S. Sisodia, R. Khandelwal, and A. Anuragi, "Categorization performance of unsupervised learning techniques for web robots sessions," in *Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*, jul 2018, pp. 1370–1374.

[42] D. S. Sisodia and N. Verma, "Performance evaluation of density-based clustering methods for categorizing web robot sessions," in *Proceedings of the 2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, dec 2018, pp. 1–5.

[43] W. Zhu, J. Qin, R. Kong, H. Lin, and Z. He, "A system framework for efficiently recognizing web crawlers," in *Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, oct 2018, pp. 1130–1133.

[44] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and Y. Kompatsiaris, "Towards a framework for detecting advanced web bots," in *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES'19)*, aug 2019, pp. 1–10.

[45] W. Zhu, H. Gao, Z. He, J. Qin, and B. Han, "A hybrid approach for recognizing web crawlers," in *Proceedings of the 14th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2019)*, 2019, pp. 507–519.

[46] R. Patil and P. Trivedi, "Determination of user navigational patterns from server log files using hadoop techniques," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 8, no. 6, pp. 1864–1870, jun 2020.

[47] S. Rovetta, G. Suchacka, and F. Masulli, "Bot recognition in a web store: an approach based on unsupervised learning," *Journal of Network and Computer Applications*, vol. 157, pp. 1–15, may 2020.

[48] G. Suchacka and J. Iwanski, "Identifying legitimate web users and bots with different traffic profiles–an information bottleneck approach," *Knowledge-Based Systems*, vol. 197, pp. 1–18, jun 2020.

[49] J. P. Barddal, H. M. Gomes, and F. Enembreck, "Sfnclassifier: A scale-free social network method to handle concept drift," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, 2014, pp. 786–791.

[50] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey," *Information Sciences*, vol. 490, pp. 344–368, 2019.

[51] C. Garcia, D. Leite, and I. Škrjanc, "Incremental missing-data imputation for evolving fuzzy granular prediction," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 10, pp. 2348–2362, 2019.