A Study of Different Web-Crawler Behaviour

Alexander Menshchikov, Antonina Komarova, Yurij Gatchin, Anatoly Korobeynikov, Nina Tishukova
Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics
Saint-Petersburg, Russia
menshikov@corp.ifmo.ru, {piter-ton, nina.tishukova, korobeynikov_a_g}@mail.ru, gatchin@mail.ifmo.ru

***Abstract*—The article deals with a study of web-crawler behaviour on different websites. A classification of web-robots, information gathering tools and their detection methods are provided. Well-known scrapers and their behaviour are analyzed on the base of large web-server log set. Experimental results demonstrate that web-robot can be distinguished from human by feature analysis. The results of the research can be used as a basis for comprehensive intrusion detection and prevention system development.**

I. Introduction

These days a problem of unauthorized massive and automated information crawling on the Internet becomes more and more serious. Website protection systems tend to be essential. Important resources and services migrate to the Internet, where they come across with a wide variety of threats such as automated information gathering by web-robots and competitive intelligence [1], [2]. In Russia e-commerce market has a strong tendency to grow. According to different research, its growth is about 15% a year. In 2016 e-commerce revenue accounted for 850 billion rubles [3]. Because of these facts, it is important to provide higher data integrity, confidentiality and availability of websites [2].

There are special tools for information gathering on the Internet. These programs called web-robots, parsers and crawlers. We can divide them into two groups according to their objectives: robots, used for legal purposes (content analysis, indexing for search systems, site mirroring etc.) and robots, used by criminals [4].

Web-robots can not only gather and process information, but also behave actively on web-resources: buy goods, write advertising posts and comments, send spam and exploit vulnerabilities. Additionally web-robots can be responsible for intensive activities resulting in high loads on web-servers, and therefore slow down website performance causing availability issues for regular users [5]. Scrapers require considerable bandwidth and usually operate in several threads during a long period of time. Poorly written crawlers can also download dynamic pages infinitely or send malformed requests to web server.

To give instructions about their site parsing to web robots, website owners use the robots.txt file, which is called “The Robots Exclusion Protocol”. About a third of web resources use this standard to regulate crawling activities [7]. Not each web robot cooperates with the standard namely email harvesters, spambots, malware, and robots, that scan the site for security vulnerabilities, as well as other malicious robots may ignore these recommendations [6].

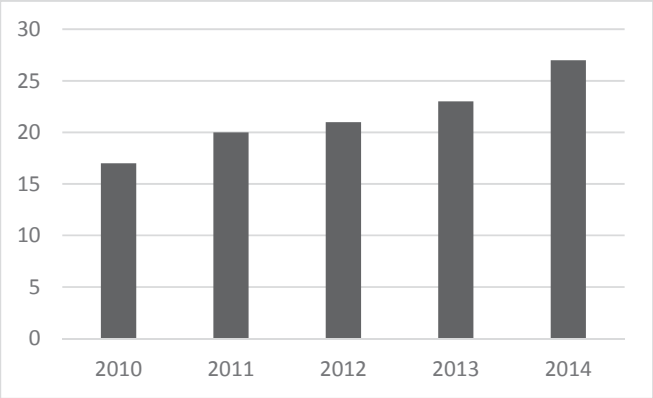

Fig. 1. Average scraping traffic per site, (%)

The amount of scraping traffic identified in 2014 increased by 17% since 2015 and by 59% compared with 2010 [2]. To disable new scraping sources it is essential to develop new methods of web-crawling detection.

Scrapers have become more aggressive and elusive, using a larger number of IP addresses to conduct their activity and to avoid detection. They also use a large amount of infected computers and other devices for their purposes.

Today, there is a necessity of complex methods that combine 24/7 monitoring and the best international practices in the field of web robots detection. Business also interested in information protection against automated data gathering, since it directly affects its profit [3].

II. Web-robot classification

One of the most interesting features of web-robots is purposefulness. Robots are commonly designed for specific purposes connected with obtaining actual information with cost minimization and gathering speed acceleration by incorrect behaviour and redundant queries exception. This behaviour is typical for both legitimate and non-ethic robots and it allows tracking connection of their behavioural templates of information processing on web-resource. In other words it allows distinguishing robot traffic from human one based on their behaviour. Robots are usually divided into three general categories [8]:

1) Amateur web-robots, which use direct web page crawling and only simple requests. These scrapers have a low amount of dedication and resources. Usually they are used by inexperienced users without large budget for information gathering.

2) Advanced web-robots, which try to act like legitimate users. They use several IP addresses and change user-agent strings and browsing methods periodically.

3) Professional web-robots, which use complicated behaviour algorithms and they are often manually tuned for each web-resource.

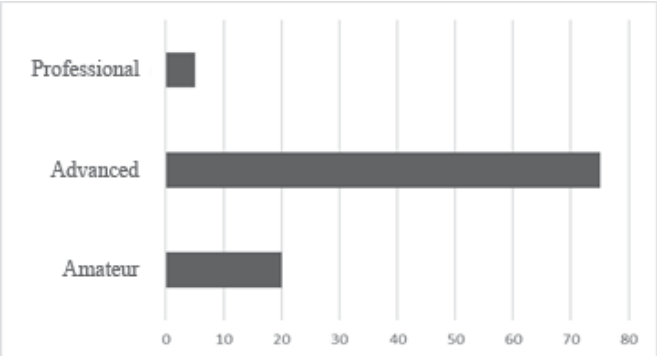

Fig. 2. Scraper types usage, (%)

Figure 2 shows the percentage of web-robots categories [2]. Advanced crawlers are the most popular ones whereas professional parsers are used significantly more rarely.

Attacks by using web robots are aimed to obtain required information on the web-site. An attacker could exploit the knowledge about software versions, web server and information about appropriate updates for related types of attacks. In addition, the information itself may comprise commercial secrets and personal data [9].

Web resources often provide more information than users need and criminals can use this fact for their purposes. Even the slightest information about system may result a full discredit [10]. Today automated information gathering tools allow attackers to collect data from different resources massively.

Crawling systems are used for competitive intelligence. Business competitors collect additional information to create their own effective system with stolen content [11].

Web robots are extremely dangerous for related to e-commerce web resources. Such resources demonstrate unique content with commercial value, for example:

- Travel
- Online Classifieds
- Online Directories
- Ticketing
- Blogs and sites with unique content
- Informational resources and libraries
- Social networks
- Other resources that contain personal data

Companies in the travel industry remain top targets for scrapers, closely followed by Online Directories and Online Classifieds [1]. The most scraped industries all share the same problem. They have a lot of publicly available data and rely on it for their business success. If competitors or other operators steal data and use it for their purposes, it will affect them negatively and in the long run be a threat to their business model [12].

III. Web-robot detection

Web-robot detection methods can be classified by their operational principles, launching strategies and using techniques. According to the first criterion robots are divided into four categories (Fig. 3). The second criterion divides them into active ones, which work during robot query, and delayed ones, which run afterwards. These techniques include filtration, machine learning methods etc.

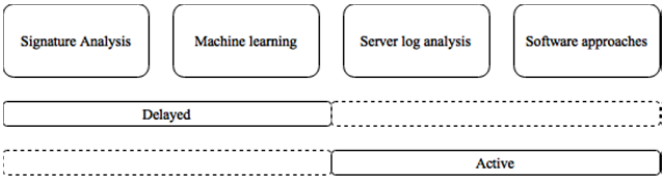

Fig. 3. Detection method classification

Real-time log analysis is a simple web server logs processing. It includes such metrics as: identification of suspicious HTTP headers, the analysis of User-Agent and Referer fields, IP filtering by country or organization. The main advantage of these methods is the implementation simplicity and data processing speed. However, it can detect only known web-robots. Log analysis is generally used for the amateur parsers detection.

Signature traffic analysis is based on the detection of certain characteristics, which are inherent for robotic systems in contrast to the human user. For example, using navigation with the same nesting level, too high query rate, downloading only HTML pages (without scripts and CSS files). This method uses the deviation of the metrics values based on a typical user behavior, as opposed to the previous method, which searches for specific patterns in the logs. The advantage of this method is in high coverage, but it needs sensitivity setting for each metric.

Machine learning methods. These methods include traffic and web-server logs analysis. It provides a statistical traffic analysis in order to detect crawlers. It usually uses metrics that are typical for signature analysis. The advantage of this method is in ability to detect previously unknown parsers, however it needs learning and training to achieve the required accuracy of detection and eliminate false positives, which can be very difficult in the manual mode.

Traps is a purely technical way to distinguish human from robot. They include Turing test, special obfuscated JavaScript functionality, invisible links, Flash applets, browser local storage, cookies, ETag, browser history, etc.

The most popular way to provide Turing test on website is CAPTCHA. A CAPTCHA is a program that can generate tests that most humans can pass, but automated information gathering tools cannot pass.

As for JavaScript protection, it works as following. Browser generate a special code (or multiple codes) and send it to the server. The JavaScript function that is responsible for the code is formed by a complex logic and it is often obfuscated.

One of the most effective ways to protect data against automated parsing - a frequent modification of the web-page layout. This can include not only a change of names and identifiers but also a change of elements hierarchy.

Some websites limit the frequency of requests and the amount of data that can be downloaded by each IP address or user.

Every technology brings new potential ways to provide user identification, tracking and crawling detection. For example, browser history and caching strategies can be used in order to detect parsers. We can show user a special generated image and check if user try to download it after. When the image changes, the checksum changes. Therefore, when the browser has the image and knows the checksum, it can send it to the webserver for verification. The webserver then checks whether the image has changed and identify user by this tag.

D. Doran and S. Gokhale [8] classified robot detection techniques into four main categories, which are closely related to those that are proposed by this article: syntactical log analysis, analytical learning techniques, traffic pattern analysis, and Turing test systems.

Junsup Lee and others [5] provide a classification of web-robots based on over one billion requests. They used workload and resource type characteristics to compare behaviour of web-robot and human.

Tan, Pang-Ning, and Vipin Kumar [11] developed an effective characterization metric, based on navigational parameters.

S. Kwon, YG. Kim and S. Cha [10] described a method of web-robots detection by analysis of typical patterns created by the sequences of request file types.

The approach proposed in this paper analyzes behavioral, timing and structural parameters simultaneously.

For the performance purposes, methods can be combined. For example, results of time-consuming analysis are used to create simple rules and filters for intrusion prevention system. The system can process a large amount of web logs and produce a simple set of IP addresses or suspicious HTTP headers, which can be easily checked by real-time detection component. Real-time filters help to clear logs from simple bots and spam traffic, which increase performance of the composite system.

IV. Crawling challenges

While developing any automated crawling system, it is essential to take into consideration various limiting factors and major challenges the developers of such programs may face with. They are:

1) The necessity of manual configuration and debugging of the system for parsing sites with a complex structure.

2) Information gathering systems have to be able to handle large amounts of data within a short period of time;

3) The design and layout of web-sites can change frequently. It affects to scraping systems and spoil parsing results. Operators have to check them regularly and fix parsing rules manually after every change.

It is important to understand the difficulties the parser developers face with and how they can be used to protect web-resources against web-robots. When we increase the cost of web-robot development, the quantity of attackers can be reduced [9].

V. Web-robot behaviour parameters

Common (legitimate) web-robot behaviour is similar to the behaviour of web-robots used by hackers. They differ in aims of information gathering, types of content, and compliance with rules and wishes of resource administrators, as it is described in the robots.txt file. Such parameters as the query source address and the HTTP User-agent header make it possible to identify the legitimate web robots and distinguish them from humans [10], [11], [12]. We have studied their characteristics, in order to detect unknown web-robots, which hide their presence on website. We use the fact that robotic behavior patterns are similar for both legitimate and unknown web robots in accordance with the problems of information collecting as described above [13].

We identify five main categories of robot behaviour characteristics [14], [15]:

1) Timing parameters based on the time intervals between requests within one or several sessions;

2) Structural parameters that depend on the HTTP packet structure and the correctness of certain fields and protocols;

3) Based on the content type;

4) Error based parameters that include the number and the proportion of errors in queries;

5) Behavioral parameters, based on the crawler actions on web-resource.

Today, web resources have dynamic structure. They often change their content every few seconds. Under these conditions, by the time a web-robot will carry out a full parsing of a resource, a part of the data will be out of date and crawler will have to start over. This is especially actual for robots that work with ads sites, e-commerce, auctions and booking resources because crawlers should obtain current information as quickly as possible.

From the standpoint of the web robots owners, the collected information should be relevant. This is also true for crawlers that belong to the search engines. To evaluate the characteristics we can use web-pages age and relevance analysis [12], [13].

Relevance (R) is a binary measure, which indicates whether the current local data is relevant, or not.

Age (A) is a measure that indicates how outdated the local copy is. Modification time (mtime) is the time of the first change of the document.

$$A = \begin{cases} 0 & \text{if local data is not modified} \\ t - mtime & \text{otherwise} \end{cases} \quad (2)$$

The automated information gathering system always has to minimize the number of outdated information in its database. This behavior can be illustrated as a system with multiple queues, and one main server.

There are many additional parameters that affect the priority of new pages loading. For example, re-visiting pages with the same frequency, regardless of their rates of change, or re-visiting more often the pages that change more frequently. The visiting frequency is directly proportional to the estimated change frequency [15], [16].

Features, which are based on timing characteristics of visits, are presented in Table I.

TABLE I. KEY WEB-ROBOTS DETECTION FEATURES BASED ON TIMING

#	Title	Description
1	totalDuration	Total session duration
2	averageInterval	Average interval between requests
3	stddevInterval	The standard deviation of the time between requests
4	cycleWeight	The measure of repeatability queries at regular intervals
5	datetimeFraction	Percentage of requests duration

It should be noted that in this context more behavioral and complex metrics are used. For example, analysis of the time intervals distribution, the frequency of queries, session activity variability by the time of day etc.

Features, which are based on behavioral characteristics of visits, are presented in Table II.

TABLE II. KEY WEB-ROBOTS DETECTION FEATURES BASED ON BEHAVIOURAL PARAMETERS

#	Title	Description
1	traversalHeight	Pages crawling depth (max, min, average, deviation)
2	traversalWeight	Number of visited pages on the site for each of the nesting levels (max, min, average, deviation)
3	cycleCount	Number of cyclic paths of different length (N>2)
4	returnsCount	The number of returns to the previous level of nesting

VI. WEB-ROBOT BEHAVIOUR STUDY

In this research, we have used a large web resource query log within the two day period. We have studied a sample containing 831 000 requests and marked 9751 as known web crawlers. We have divided them into 413 independent

$$R = \begin{cases} 1 & \textit{if local data is equal to remote} \\ 0 & \textit{otherwise} \end{cases} \quad (1)$$

sessions. Sessions related to well-known web robots have been determined by analyzing the following features: the fact of referring to robots.txt, IP address, subnet, User-agent. The period of life for the session division process has been set to 30 minutes. Session identification algorithm can be represented by the following approximate pseudo-code (algorithm 1).

Algorithm 1 Session identification algorithm

```
for request in Requests:
    for session in ActiveSessions:
        if (request.time - session.lastTime > delta):
            session.close()
        else:
            if (session.containsIP(request.ip) and \\
session.containsUserAgent(request.userAgent)):
                session.add(request)
            else:
                newSession = new Session()
                newSession.add(request)
```

This problem occurs if there is no possibility to influence to the web-resource infrastructure [14]. To solve this problem, we have to solve the sessions identification problem and analyze web server logs after that.

To do this, we have to analyze web server logs and solve the sessions identification problem. If we can change web-server settings than users can be divided into sessions by assigning unique values in Cookies.

We select structural parameters to study the behavior of web robots. The classified characteristics are presented in Table III.

TABLE III. KEY WEB-ROBOTS DETECTION FEATURES BASED ON HTTP CONTENT

#	Title	Description
1	totalPages	Total requests number.
2	nonStaticRequests	Count of requests to web page.
3	staticRequests	Count of requests to static files and multimedia content: .css, .js, .jpg, .png, .gif, .pdf.
4	robotsTXTRequest	Queries to robots.txt file.
5	errorCodes3xx	Count of 3XX error during session.
6	errorCodes4xx	Count of 4XX error during session.
7	HEADRequests	HTTP HEAD requests count.
8	imagesCount	Count of requests to images .png, .jpg, .gif.
9	scriptsCount	Count of requests to file with .css, .js extensions.
10	unassignedReferer	Count of requests with empty or «-» referrer.

The logs were pre-processed to remove humans sessions and unknown web robots. Queries that remained in the results were divided into sessions for which the behavior characteristics were calculated. Then, irrelevant and uninformative entries were excluded from the sessions. As a result of a comparison, 56 sessions were produced.

Web robots features were compared with those of the humans. This comparison demonstrated a significant difference. The most significant in the context of the structural characteristics of the relations were types of files. Figure 4 shows the dependence of the requests for static files percentage from the session length for known crawlers and ordinary users. It is worth noting that several malicious web robots were found among ordinary users.

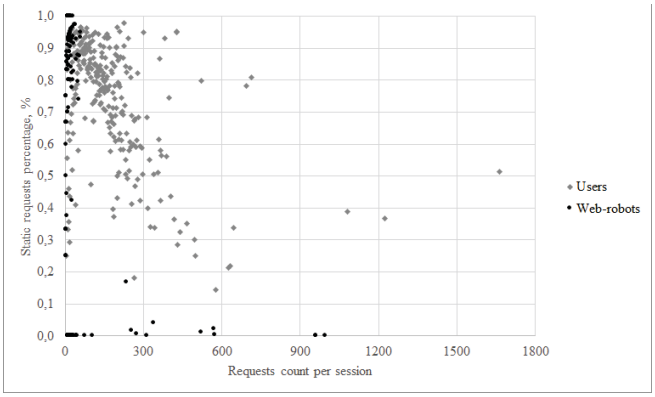

Fig. 4. Dependence of static requests percentage from session length

It can be concluded that the web robots detection methods based on structural features can detect certain types of crawlers, but do not cover the whole variety of crawlers, unlike the time-based and behavioral methods. This issue requires further research.

There are also some advanced structural metrics include number of bytes exchanged between server and client, page popularity, workload analysis [15]. However, we decided to took them to a behavioral category.

VII. Results

We developed a system and computer program that analyzes the behavior of web robots in automatic mode to study their features and produce patterns for detection of malicious visits, as well as for blocking explicit web robots in online mode. An exemplary architecture of the system (Figure 5) consists of five modules:

1) Query processor – a component that receives behavior parameters from the web server;

2) Sessions analyzer – a component that classifies sessions as human and robotic;

3) Component that calculates behavior characteristics for each session;

4) Decision-making component that uses decision trees and features thresholds;

5) Results analyzer – a component representing the characteristics, which helps system operator to edit features and conduct experiments.

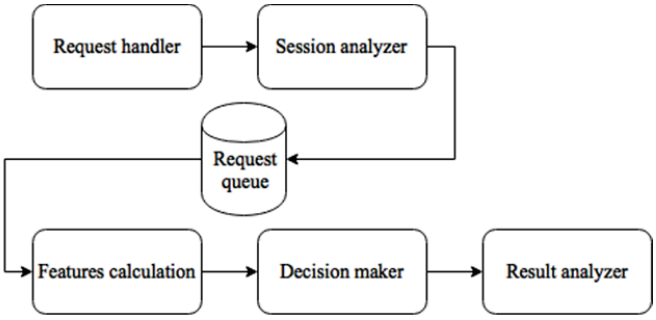

Fig. 5. Web-robots behaviour analyzing system

The features calculation process run logs sanitizing procedure before start. The procedure can be illustrated by the following pseudo-code.

Algorithm 2 Logs sanitizing procedure

```
def clear_from_common_crawlers (log_list_data):
    return filter (
    lambda x: not BOTREGEX.search(x.user_agent),
    log_list_data)
def filter_by_ips (log_list_data, white_list=set(),
    black_list=set()):
    if len(white_list):
        log_list_data = filter(lambda x: x in
        white_list, log_list_data)
    return log_list_data
def get_user_agents_by_ips (log_list_data, ips):
    results = {}
    for ip in ips:
        results[ip] = next(obj for obj in log_list_data if
        obj == ip)
  return results
```

The prototype was tested on a sample subset of web server logs containing 200,000 queries and 39,635 traffic sources.

Our system found 5347 different versions of web browsers. For each version, query rates for every IP address were calculated.

Twenty sources have been marked as robotic based on the requests frequency table analysis.

File types analysis helped us to find another ten web-robots sources.

Referring table was useless on this experiment; we could eliminate only those requests that did not have any referrer.

More than 300 browsers have been marked as suspicious by User-Agent analysis. For example, «AOLserver-Tcl / 3.3.1 + ad13», which was found in the logs again in 1862 with a single IP address.

Several web-robots were find by error (3xx, 4xx) analysis and standard directories automated lookup (for example, admin directories).

With the use of this system, new features have been designed and refined to improve web-robots detection. For each of the features under consideration a set of thresholds for better classification was designed. The data was divided into training set, which was formed on the basis of threshold values, and the test set that has been marked up manually.

The results showed the classification accuracy at 0.83 and precision at 0.92. These values may be specified in the study of a larger set of data, taking into account all detection categories. A classification using machine-learning methods will be the subject of further research.

VII. Conclusion

The web-robots detection problem requires a whole range of tools. Firstly, web-robots detection methods based on certain parameters and information about their activity. Secondly, a system that helps with the use of these methods, gathers all the necessary information to carry out its preprocessing, processing and decision-making. Third, the framework to adjust the detection system and monitoring of its operations.

The significance of the results is in new methodological approaches and developed tools. They can be used to protect web resources from automated information gathering. We studied a set of web server logs and found robotic sources by comparing the characteristics of the visitor's behavior. The results allow automatic detection of web robots activity on website and disabling their sources.

This study will serve as a stepping-stone for the construction of an integrated approach to ensure the security of web-resources and for the generation of representative data sets that are necessary for machine learning methods applied to the problem of web-robots detection.

Acknowledgment

This research is conducted within the framework of the 615878 research project "security methods and systems". We would like to thank Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics for financial support. We would like to thank Dean of Information Security and Computer Technologies Faculty Bobtsov A. A. We also express our gratitude to all staff of the Computer System Design and Security Department for scientific support.

References

[1] Report East-West Digital News [Electronic resource] – Mode of access: http://www.ewdn.com/files/ecom-rus-download.pdf/, free (date accessed: 27.10.2016).

[2] Report of the company scrapesentry [Electronic resource] – Mode of access: https://www.scrapesentry.com/scrapesentry-scraping-threat-report-2015/, free (date accessed: 27. 10.2016).

[3] Report of the Association of companies of Internet trade [Electronic resource] – Mode of access: http://www.akit.ru/wp-content/uploads/2016/05/E-commerce_1Q2016-FINAL.pdf free (reference date: 01.11.2016).

[4] Menshchikov A.A., Gatchin YU.A. Metody obnaruzheniya avtomatizirovannogo sbora informacii s veb-resursov // Kibernetika i programmirovanie [Cybernetics and programming]. – 2015. – № 5. – S. 136-157

[5] Junsup Lee, Sungdeok Cha, Dongkun Lee, Hyungkyu Lee, Classification of web robots: An empirical study based on over one billion requests // Computers & Security. – 2009. – V. 28. – № 8. – P. 795-802.

[6] Robots Exclusion Protocol Guide [Electronic resource]. – Mode of access: http://www.bruceclay.com/seo/robots-exclusion-guide.pdf free (reference date: 01.11.2016).

[7] Sun, Yang, Isaac G. Councill, and C. Lee Giles. "The ethicality of web crawlers." Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Vol. 1. IEEE, 2010.

[8] D. Derek, S. Gokhale A Classification Framework for Web Robots // Journal of American Society of Information Science and Technology. – 2012. – V. 63. – P. 2549–2554.

[9] G. Jacob, E. Kirda, C. Kruegel, G. Vigna PUB CRAWL: Protecting Users and Businesses from CRAWLers // Proceeding Security'12 Proceedings of the 21st USENIX conference on Security symposium. – 2012. – P. 25–36.

[10] S. Kwon, YG. Kim, S. Cha Web robot detection based on pattern-matching technique // Journal of Information Science. – 2012. – V. 38(2). – P. 118–126.

[11] Tan, Pang-Ning, and Vipin Kumar Discovery of web robot sessions based on their navigational patterns // Intelligent Technologies for Information Analysis. Springer Berlin Heidelberg. – 2004. – P. 193-222.

[12] Stassopoulou, Athena, and Marios D. Dikaiakos Web robot detection: A probabilistic reasoning approach // Computer Networks. – V. 53. – № 3. – 2009. – P. 265-278.

[13] Lu, Wei-Zhou, and Shun-zheng Yu Web robot detection based on hidden Markov model // 2006 International Conference on Communications, Circuits and Systems. – 2006.

[14] T. H. Sardar, Z. Ansari Detection and Confirmation of Web Robot Requests for Cleaning the Voluminous Web Log Data // Proceeding International Conference on the IMpact of E-Technology on US. – 2014. – V. 28. – P. 795–802.

[15] D. S. Sisodia, S. Verma, 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. – 2015. – V. 3. – P. 1–10.

[16] Ferrara, Emilio, et al. Web data extraction, applications and techniques: A survey // Knowledge-Based Systems. – 2014. – V. 70 – P. 301-323.

[17] Menshchikov A.A., Gatchin YU.A. Postroenie sistemy obnaruzheniya avtomatizirovannogo sbora informacii s veb-resursov // Inzhenernye kadry - budushchee innovacionnoj ehkonomiki Rossii: Materialy Vserossijskoj studencheskoj konferencii: v 8 ch. – 2015. – T. 4. – S. 58-61

[18] May, Wolfgang, and Georg Lausen. A uniform framework for integration of information from the web // Information Systems. – 2004. – V. 29. – № 1. – P. 59-91.

[19] DS. Sisodia, S. Verma, OP. 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. – 2015. – V. 3. – P. 1–10.

[20] Guo, Weigang, Shiguang Ju, and Yi Gu. Web robot detection techniques based on statistics of their requested URL resources // Computer Supported Cooperative Work in Design, 2005. Proceedings of the Ninth International Conference. – V. 1. – 2005.

[21] Geens, Nick, Johan Huysmans, and Jan Vanthienen. Evaluation of Web robot discovery techniques: a benchmarking study // Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining. Springer Berlin Heidelberg. – 2006. – P. 121-130.

[22] D. Derek, S. Gokhale Discovering New Trends in Web Robot Traffic Through Functional Classification // Proc. IEEE International Symposium on Network Computing and Applications. – 2008. – P. 275–278.

[23] D. Derek, S. Gokhale Discovering New Trends in Web Robot Traffic Through Functional Classification // Proc. IEEE International Symposium on Network Computing and Applications. – 2008. – P. 275–278.

[24] B. Quan, X. Gang, Z. Yong, H. Longtao Analysis and Detection of Bogus Behavior in Web Crawler Measurement // Procedia Computer Science. – 2014. – V. 31. – P. 1084–1091.
[25] D. Derek, S. Gokhale Detecting Web Robots Using Resource Request Patterns // Procceeding 11th International Conference on Machine Learning and Applications. – 2012. – V. 1. – P. 7–12.
[26] Chen, Hsinchun, and Michael Chau. Web mining: Machine learning for web applications // Annual review of information science and technology. – 2004. – V. 38. – P. 289-330.