ICIC Express Letters
Volume **13**, Number **12**, December **2019**

ICIC International ©2019 ISSN 1881-803X
pp. 1149–1156

HUMANS AND BOTS WEB SESSION IDENTIFICATION USING K-MEANS CLUSTERING

Muhammad Medhat[1], Yasser Fouad Hassan[1,2] and Ashraf Elsayed[1,3]

[1]Department of Mathematics and Computer Science
Faculty of Science
Alexandria University
Baghdad, Al Bab Al Gadid WA Mansha, Qism Moharram Bek Alexandria Governorate 21568, Egypt
muhammad.medhat@alexu.edu.eg

[2]Faculty of Computer Science and Artificial Intelligence
Pharos University
Alexandria 21648, Egypt
y.fouad@pua.edu.eg

[3]University of Science and Technology
Zewail City of Science and Technology
October Gardens, 6th of October, Giza 12578, Egypt

Received April 2019; accepted July 2019

Abstract. *Denial of Service (DoS) attacks are one of the most damaging attacks on the Internet security today. This is done by forcing some computers to perform several disturbing tasks to make the machine or network resource unavailable to its intended users. Many major companies have been the focus of it. Because this attack can be easily engineered from nearly any location, finding those responsible can be extremely difficult. Since the web crawlers have a big share on browsing the Internet websites recently, that makes the target subject to being attacked by them. This study presents an analysis for a web log file that records all server requests during the time span of 30 June 2018 to 30 July 2018 obtained from a popular university website to generate the web sessions during this period to easily identify sessions behaviors, and then the K-means algorithm is used for clustering the output of this analysis, based on these behaviors. This will help us in using the assumptions set on this work to allow labeling for each cluster to human visitors, benign crawlers, malicious crawlers and unknown requests.*

Keywords: Web usage mining, Data mining, Clustering, K-means

1. **Introduction.** The growth of massive amount of Internet websites and online services increased in the last few years, resulted in a large amount of web applications, devices to use them; however, the vulnerability of the web architecture can provide an opportunity for various attacks on the security of web based applications. There has been the growing need to develop advanced tools for retrieving information on the web content, structure, and usage. Such tools are web bots (also called web robots, spiders, or crawlers). They can traverse the web by following the structure of hyperlinks, collect different kinds of information, and perform specific tasks on websites. Examples of robots are stated in various works like in [1, 2, 3]. This means that malicious crawlers can infiltrate and send a flood of messages to servers, which is a common way of penetrating systems called Denial of Service (DoS) [4]. The most common way of conducting a (DoS) attack is by sending a flood of messages to the target (e.g., a machine hosting a web site) with the aim to interfere with the target's operation, and make it hang, crash, reboot, or do useless work [5]. Analysis for web traffic is vital, to identify visitors for website. Data mining [6, 7] is most suitable for this need. It analyzes databases and classifies/partitions it so that

DOI: 10.24507/icicel.13.12.1149

any organization takes decision based on this classification and can improve their future plans. There are many techniques of data mining by which we can detect the hidden patterns in the databases. These techniques fall under two categorie, i.e., supervised and unsupervised learning.

1.1. **Bot share in overall web server traffic.** According to the results of earlier analyses for e-business workloads the share of robot requests has differed from several (3.2% in [8]) to a dozen or so (15% in [9], 16% in [10]).

1.2. **Motivations.** Numerous amount of attacks are being done recently by vulnerable visitors against the servers or web applications. The web application architecture cannot differentiate between the human visitor or the bot visitor, so we started making a research and checking the sessions statistics for visitors by extracting the useful information for each user session and used the data mining techniques to cluster the extracted sessions file. This paper is focused on using the K-means technique for clustering the web sessions generated from the provided log file.

The paper is organized as follows. The rest of the current section briefly describes some related work. Part 2 presents the web server log data underlying our research and the research methodology. Part 3 illustrates the experiment done in this paper and compares key characteristics of bot and human sessions. Part 4 discusses the work done in the paper, then it is concluded in Part 5 and suggests prospective future work.

1.3. **Related work.** In this section the most related researches done regarding clustering, access log analysis have been reviewed. A novel semi-supervised method to detect spam queries in search engine is perposed in [11], the activity log of a popular Iranian local search engine is used in the experaments, they focused on detecting queries in real time rather than using the offline methods.

K-means and threshold based methods are evaluated on synthetic data with popular validity measures and validity indices in [12]. The experimentation carried out in this work uncovers that threshold based algorithm performs better than K-means algorithm for popular validity measures and validity indices. A proposed clustering method called a sample-based hierarchical adaptive K-means (SHAKM) [13], utilizes the adaptive K-means clustering algorithm to determine the correct number of clusters and to construct an unbalanced tree for a large scale video retrieval. Session Anomaly Detection (SAD) is a method developed in [14] as a Bayesian estimation technique. In this model, web sessions are extracted from web logs and are labeled as 'normal' or 'abnormal' depending on whether it is below or above the assigned threshold value. In addition, two parameters that are page sequences and their frequency are investigated in training data. In order to test their results, the authors used Whisker v1.4 as a tool for generating anomalous web requests and it is asserted that the Bayesian estimation technique has been successful for detecting 91% of all anomalous requests. Therefore, two points making this article different from the others are that SAD can be customized by choosing site-dependent parameters; and the false positive rate gets lower with web topology information.

Analysis of two web based attacks which are i-frame injection attacks and buffer overflow attacks is done in [15]. For analysis, log files created after attacks are used. They compare the size of the transferred data and the length of input parameters for normal and malicious HTTP requests. As a result, they just have carried out descriptive statistics and have not mentioned any detection techniques. Analyzing for users behaviors is done in [16], where the authors performed several surveys to be available for the academicians and the other researchers for different opportunities, predictions for user intentions are not recorded in the log file.

2. Research Methodology.

2.1. Web server log data description.

When an Internet user visits a website, their web browser (which is a web client) communicates via the HTTP protocol with the server hosting the site. For each web page requested by the user, their client typically issues a series of HTTP requests to the server: one request for a page description file and the following requests for objects embedded in the page, such as images or video files. After receiving HTTP responses the client assembles the page and displays it in a browser window. A web client may represent not only a human user but it may also be a computer program, i.e., a web bot. Data concerning each incoming HTTP request is recorded in the access log file stored at the web server.

2.2. Reconstruction and characterization of user sessions.

Based on HTTP requests user sessions were reconstructed. A user session means a sequence of requests issued by a web client during the single visit to the web application. Each individual user was identified based on two data fields describing HTTP requests: the client IP address and the user agent field. Consecutive user sessions were reconstructed based on the requests' timestamps, assuming a minimum 30-minute interval between two subsequent sessions of a given user (the value of 30 minutes has been commonly applied in previous web traffic analyses, e.g., in [5, 17]). Afterwards, each user session was described with a number of attributes:

- Session ID is a composite value to help the software in identifying sessions database.
- Session Time – The date time of the first request in the session.
- Session Length – A numerical value that represents the requests number in the session.
- Session Duration – A time span witch is the difference between first and last request date time, for sessions containing more than one page.
- Mean Time per Page $\bar{T}_j$ for the session j where mean time can be obtained by finding the time difference between every two successive requests.
- Session Data Size – A numerical value to represent the sum of all of the bytes transferred in each request.
- User Agent – A string represents the user agent used for the session.

We decided to compute the aforementioned attributes because some previous user session analyses for different environments [5, 18, 19] reported that these session features may be useful in distinguishing web robots from human users.

2.2.1. *Identification of bot sessions.*

There are few ways to identify sessions of web robots. First, one should check if the file "robots.txt" was accessed in a session. Cooperative robots should request this file at the beginning of each visit to a site in order to read which parts of the site they can access. Second, "ethical" bots should inform a web server about their identities via their user agent fields, containing the name of the robots that are listed in an online database in [20]. Moreover, some robots not included in these databases were identified based on keywords contained in user agent fields ("bot", "spider", "crawler", "worm", "search", "track", "harvest", "dig", "hack", "trap", "archive", or "scrap"), as well as through a semi-automatic inspection of user agent fields.

In practice, not all robots access the file "robots.txt" or declare their identities in user agent fields. However, some of such bots may be still identified based on the character of their interaction with the site, which proceeds differently from the interaction of human users. Humans usually communicate with the site via the web interface and follow navigation paths according to the site topology. Each web page request is typically followed by a group of requests for embedded objects (usually images). Moreover, the successive page requests are separated with some time intervals called "user think times". In contrast, robots tend to reveal navigational patterns incompatible with the site topology and have un-intuitive session characteristics, e.g., the extremely low mean time per page.

3. Experimental Design.

3.1. Training data.

Alexandria University (http://www.alexu.edu.eg) server logs are used in the work of this paper. This website is a public website and also has a content specific information that requires a user to login for it.

3.2. Experiment overview.

The log file has to be processed before working to analyze and obtain its output sessions file according to the steps stated in Section 3.3. Then sessions file will be clustered with WEKA software using the K-means algorithm. It will produce the clusters for web requests that can be labeled later according to conditions in Section 4.1.

3.3. Pre-processing.

3.3.1. *Data description.* The used log dataset includes 3974251 lines with useful information as displayed in Figure 1, and it is used as an input for the software to generate a list of vectors with 9 attributes named as the following.

- Request IP which is the IP address of the remote host visiting the website,
- Date and time of the request arrival at the server,
- HTTP method (e.g., GET, HEAD, POST) used in the request,
- URI identifying the requested server resource,
- Version of the HTTP protocol (HTTP/1.0 or HTTP/1.1),
- HTTP status code [21],
- Data size transferred between the client and server during the request,
- Referrer URL – URL which had referred the user to the site via a hyperlink or "–" if a referrer was not given,
- User-Agent – a string defines a description for the client Internet browser and operating system or "–" if the information was not given.

Each line in the log file is represented as one vector and they will be used in the step in Section 3.3.2.

```
40.77.167.97 - - [11/Oct/2017:16:58:19 +0200] "GET /robots.txt HTTP/1.1" 200 842 "-"
        "Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)"
```

Figure 1. Sample log line

This line describes a request sent by a web client with the IP address 40.77.167.97, whose user identifier is not available (displayed as "–"). The request was served 12 October 2017 at 10:29:03 (according to +2 hours Greenwich Time) and it concerned downloading (by using the GET method) an the robots.txt file identified by URI "/robots.txt". The request was successfully served (a status code is 200) and the server sent to the client 842 bytes in response. A referrer field is unassigned. The client was Mozilla 5.0 which used the protocol HTTP/1.1. One can notice that the user was not a human but Bing's web crawling bot (the user agent field contains the bot's name, "bingbot").

3.3.2. *Session identification.* Web session is a sequence of successive web requests from the user during a single visit [22]. The software extracts the generated vectors then groups them into sessions. Session identification is the process of grouping log requests into a user session based on a pre-defined key. The software extracted all the HTTP requests from the log file and obtained the following data attributes for each session. From previous studies on web session analysis, namely [19, 23, 24], different fetchers have been adopted that are shown to be useful in identifying and distinguishing between automated and human visitors to a web site. These features are enlisted in Section 4.1.

3.4. **K-means.**

K-means [25, 26] clustering is one of the simplest and popular unsupervised machine learning algorithms. The power of K-means algorithm is due to its computational efficiency and the nature of ease at which it can be used. It is based on partitioning methodology, and it partitions n data items into k-groups where k indicates the number of clusters specified by a user. Clustering is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. For calculating distance between an item and the centroid, K-means algorithm uses the Euclidean distance measurement to minimize the objective function as in Equation (1).

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left|\left| x_i^{(j)} - c_j \right|\right|^2 \tag{1}$$

where $\left|\left| x_i^{(j)} - c_j \right|\right|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j is an indicator of the distance of the n data points from their respective cluster centroids. Algotithm 3.1 represents the basic steps for K-means clustering.

Algorithm 3.1. *K-means clustering algorithm*

```
1) Select clusters number K.
2) Randomly select K centroids.
3) Iterate until stable (= no object move group)
        Determine the distance of each object to the centroids Equation (1).
        Group the object based on minimum distance and update the centroids.
```

3.5. **WEKA.**

WEKA [27] is a collection of machine learning algorithms and data preprocessing tools. It provides the extensive support for the whole process of experimental data mining, including the preprocessing of the data for the input, classification, clustering, association rules, evaluating learning schemes statistically and visualizing the input data and the result of learning. The system is distributed under the terms of the GNU General Public License. WEKA interface has four main components.

1) Simple CLI: it provides command line interface and allows the direct execution of WEKA commands.
2) Explorer: it is an environment for exploring the data.
3) Experimenter: it is an environment for conducting experiments and to perform statistical analysis between different learning schemes.
4) Knowledge Flow: it is the Java Beans based interface for setting up and running machine learning experiments.

4. **Discussion.**

4.1. **Fetchers to identify robots.**

The features for identifying robots sessions and human sessions are enlisted in the following few lines.

Click Rate is a numerical attribute that represents the number of requests sent during the session. The click rate metric appears to be useful in detecting the presence of the web crawlers because higher click rate can only be achieved by an automated script (such as a web robot) and is usually very low for a human visitor.

HTML-to-Image Ratio is a numerical attribute of the ratio between the HTML pages and number of image requests in the single session. Web crawlers generally request mostly HTML pages and ignore images on the site which implies that HTML-to-Image ratio would be higher for web crawlers than for human users.

Percentage of 4xx error responses is a numerical attribute calculated as the percentage of erroneous HTTP requests sent in a single session. Crawlers typically would have higher rate of erroneous request since they have higher chance of requesting outdated or deleted pages.

Percentage of HTTP requests of type HEAD is a numerical attribute calculated as percentage of requests of HTTP type HEAD sent in a single session. Most web crawlers, in order to reduce the amount of data requested from a site, employ the HEAD method when requesting a web page. On the other hand, requests coming from a human user browsing a web site via browsers are, by default, of type GET.

Percentage of requests with unassigned referrers is a numerical attribute calculated as the percentage of blank or unassigned referrer fields set by a user in a single session. Typically, web crawlers would initiate HTTP requests with unassigned referrer field. Robots.txt file request witch is a nominal attribute with values of either 1 or 0, indicating whether 'robots.txt' file was requested or not requested by a user during a session, respectively. Web administrators, through the Robots Exclusion Protocol, use a special-format file called robots.txt to indicate to visiting robots which parts of their sites should not be visited by the robot. For example, when a robot visits a web-site, say alexu.edu.eg, it should first check for alexu.edu.eg/robots.txt. It is unlikely, that any human would check for this file, since there is no link from the website to this file, nor are (most) users aware of its existence.

4.2. **Dataset clustering.** Once the training data-set (comprising feature-vector representations) is generated, the log analyzer records each feature-vector in the sessions dataset to be ready for clustering using K-means algorithm. In our work we will cluster sessions into 4 clusters; we will assign clusters for human visitors, benign web crawlers, malicious crawlers and unknown visitors.

4.2.1. *Apply K-means results.*

4.2.1.1. *Sessions preprocessing.* Before applying the obtained sessions database we will manually preprocess it; we will perform a data cleaning for the obtained sessions database because it might be an error from the data source or in the log analyzer.

- Records having the attribute "Data transferred" = 0 will be removed manually.
- No instance found in the data that has a value for the attribute "Head Requests".
- Some sessions contained no web page request and only one request for an image file (such a situation is often connected with displaying a banner advertisement of the store on another web page). As these sessions cannot be regarded as intended visits to the website, we did not take them into consideration in our analysis.

4.2.1.2. *Results discussion.* A dataset of sessions is resulted after performing the step from Section 3.3.2, The resulted sessions list was used as an input for WEKA, the dataset has 155252 session instances with 9 attributes, then they are classified into 4 clusters (i.e., $k = 4$), it will be stable after 4 iterations, and then WEKA computed the percentage of instances falling in each cluster. Our clustering produced by K-means process started by initializing 4 random centroids as in Table 2, the results showed 45.7% (70990 instances) in cluster 0, 12.8% (19938 instances) in cluster 1, 2.8% (4361 in instances) in cluster 2 and 38.6% (59963 instances) in cluster 3. The sum of squared error within cluster is 324341.6519381607. Detailed results are displayed in Table 1. Figure 2 shows the final graph representations after clustering the sessions dataset obtained from the output of log analyzer.

Table 1. Final cluster centroids

Attribute	Full Data (155252.0)	Cluster 0 (70990.0)	Cluster 1 (19938.0)	Cluster 2 (4361.0)	Cluster 3 (59963.0)
IP	46.161.9.61	6.249.93.31	46.161.9.61	17.58.97.211	46.161.9.61
Time	6:43:04	6:43:16	6:43:04	8:27:23	9:21:46
Daylight	PM	PM	PM	PM	AM
NumRequests	25.5713	33.5315	3.8718	15.2165	24.1156
NumPages	5.277	6.2328	0.6348	13.9195	5.0604
NumImages	21.126	28.3949	3.4859	1.3206	19.8261
RatioImages	0.205	0.0018	0.9536	0	0.2116
NumHead	0	0	0	0	0
TotalBytes	2384759.4107	135064.0923	409760.1071	859465.6602	2264105.2209
404Ratio	0.0212	0.0161	0.031	0	0.0254
HasRobots	False	False	False	True	False

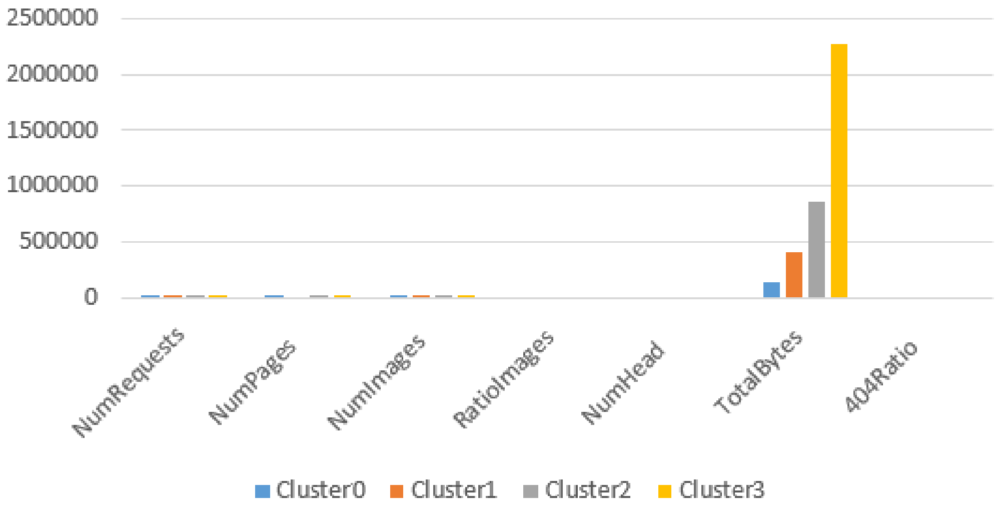

Figure 2. (color online) Clustering results

Table 2. Initial starting points (random)

Cluster 0	41.42.26.58,6:32:12,PM,45,10,36,0,0,1728265,0, 'False'
Cluster 1	196.153.3.50,3:52:36,PM,1,1,0,0,0,89725,0, 'False'
Cluster 2	17.58.97.211,11:12:58,PM,8,3,5,0,0,90834,0, 'True'
Cluster 3	47.54.218.235,4:55:53,AM,1,1,0,0,0,91832,0, 'False'

5. **Conclusion.** The paper discusses key characteristics of sessions realized by web robots and human users on the website.

In this study a custom log analyzer software is developed then used for generating sessions from the clickstream file. Then fetchers to identify robots requests were briefly discussed in Section 4.1. The output session's dataset has been preprocessed manually then used as an input for WEKA. Then K-means clustering is performed setting $k = 4$. The clustering results are illustrated in Section 4.2.1.2. The clusters obtained from WEKA output can then be classified using a supervised learning technique. As future work the log analyzer software will be modified to label the sessions into (human visitors, benign web crawlers, malicious web crawlers, and unknown visitors) according to the assumptions stated in Section 4.1.

REFERENCES

[1] S. Gianvecchio, M. Xie, Z. Wu and H. Wang, Humans and bots in Internet chat: Measurement, analysis, and automated classification, *IEEE/ACM Trans. Networking*, vol.19, no.5, pp.1557-1571, 2011.

[2] P. Hayati, V. Potdar, K. Chai and A. Talevski, Web spambot detection based on web navigation behaviour, *The 24th IEEE International Conference on Advanced Information Networking and Applications*, pp.797-803, 2010.

[3] A. Schmitz, O. Yanenko and M. Hebing, Identifying artificial actors in e-dating: A probabilistic segmentation based on interactional pattern analysis, *Challenges at the Interface of Data Analysis, Computer Science, and Optimization*, pp.319-327, 2012.

[4] N. Lyamin, D. Kleyko, Q. Delooz and A. Vinel, Real-time jamming DoS detection in safety-critical V2V C-ITS using data mining, *IEEE Communications Letters*, 2019.

[5] D. Stevanovic, N. Vlajic, A. An, D. Stevanovic, N. Vlajic and A. An, Unsupervised clustering of web sessions to detect malicious and non-malicious website users, *Procedia Computer Science*, vol.5, pp.123-131, 2011.

[6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, Diane Cerra, 2006.

[7] I. H. Witten, E. Frank, M. A. Hall and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2016.

[8] D. Doran and S. S. Gokhale, Long range dependence (LRD) in the arrival process of web robots, *International Conference on Intelligent Network and Computing*, 2012.

[9] N. Poggi, D. Carrera, R. Gavalda, E. Ayguadé and J. Torres, A methodology for the evaluation of high response time on e-commerce users and sales, *Information Systems Frontiers*, vol.16, no.5, pp.867-885, 2014.

[10] D. Menasce, F. Ribeiro, V. Almeida et al., In search of invariants for e-business workloads, *Proc. of the 2nd ACM Conference on Electronic Commerce*, pp.56-65, 2000.

[11] T. Shakiba, S. Zarifzadeh and V. Derhami, Spam query detection using stream clustering, *World Wide Web*, vol.21, pp.557-572, 2018.

[12] M. Mittal, R. K. Sharma and V. P. Singh, Validation of K-means and threshold based clustering method, *International Journal of Advancements in Technology*, vol.5, no.2, pp.153-160, 2014.

[13] K. Liao, G. Liu, L. Xiao and C. Liu, A sample-based hierarchical adaptive K-means clustering method, *Knowledge-Based Systems*, vol.49, pp.123-133, 2013.

[14] S. Cho and S. Cha, SAD: Web session anomaly detection based on parameter estimation, *Computers & Security*, vol.23, no.4, pp.312-319, 2004.

[15] A. Razzaq, Z. Anwar, H. F. Ahmad, K. Latif and F. Munir, Ontology for attack detection: An intelligent approach to web application security, *Computers & Security*, vol.45, pp.124-146, 2014.

[16] R. Chinnaiyan and V. Ilango, Analyzing the user behaviours by mining web access log files, *International Journal of Advanced Studies in Computers, Science and Engineering*, vol.4, no.11, pp.7-14, 2015.

[17] D. Doran and S. S. Gokhale, Searching for heavy tails in web robot traffic, *The 7th International Conference on the Quantitative Evaluation of Systems*, pp.282-291, 2010.

[18] A. Balla, A. Stassopoulou and M. D. Dikaiakos, Real-time web crawler detection, *The 18th International Conference on Telecommunications*, pp.428-432, 2011.

[19] M. D. Dikaiakos and A. Stassopoulou, Web robot detection: A probabilistic reasoning approach, *Computer Networks*, vol.53, pp.256-278, 2009.

[20] *The Web Robots Page*, http://www.robotstxt.org, 2007.

[21] *Mozilla*, Http response status codes | mdn, https://developer.mozilla.org/en-US/docs/Web/HTTP/Status, 1998.

[22] G. Chodak and G. Suchacka, Characterizing web sessions of e-customers interested in traditional and innovative products, *The 30th European Conference on Modelling and Simulation*, 2016.

[23] W. Gaul, C. Bomhardt and L. Schmidt-Thieme, Web robot detection – Preprocessing web logfiles for robot detection, in *New Developments in Classification and Data Analysis*, Springer, 2005.

[24] P.-N. Tan and V. Kumar, Discovery of web robot sessions based on their navigational patterns, *Intelligent Technologies for Information Analysis*, pp.193-222, 2004.

[25] S. Jain, M. A. Aalam and M. N. Doja, K-means clustering using weka interface, *Proc. of the 4th National Conference*, pp.25-26, 2010.

[26] J. MacQueen et al., Some methods for classification and analysis of multivariate observations, *Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp.281-297, 1967.

[27] I. H. Witten and E. Frank, WEKA, *Machine Learning Algorithms in Java*, pp.265-320, 2000.