TIME SERIES CLUSTERING WITH DIFFERENT DISTANCE MEASURES TO TELL WEB BOTS AND HUMANS APART

Grażyna Suchacka
Institute of Informatics
University of Opole
ul. Oleska 48
45-052 Opole, Poland
E-mail: gsuchacka@uni.opole.pl

KEYWORDS
Internet robot, Web bot, Web bot detection, Web session, Time series, Unsupervised classification, Clustering, Distance measure, Similarity measure

ABSTRACT

The paper deals with the problem of differentiating Web sessions of bots and human users by observing some characteristics of their traffic at the Web server input. We propose an approach to cluster bots' and humans' sessions represented as time series. First, sessions are expressed as sequences of HTTP requests coming to the server at specific timestamps; then, they are pre-preprocessed to form time series of limited length. Time series are clustered and the clustering performance is evaluated in terms of the ability to partition bots and humans into separate clusters. The proposed approach is applied to real server log data and validated with the use of different time series distance measures and clustering algorithms. Results show that the choice of a distance measure and a clustering method significantly affects clustering efficiency. The best results for the considered scenario were achieved for distance measures based on nonparametric spectral estimators and the Euclidean distance with a complexity correction factor.

INTRODUCTION

The share of automatically generated traffic in total Web traffic has been constantly growing for many years (Bad Bot Report 2021). To cope with the presence of artificial Web agents (robots, bots) and with possible negative consequences of their activities, many studies on bot detection have been conducted. In particular, approaches based on machine learning (ML) methods have proven to be effective in distinguishing between bots and humans (i.e., human-operated Web browsers).

The majority of approaches to detect bots on Web servers have involved LM methods in the offline scenario, i.e. for completed user visits (*Web sessions*). Primarily, supervised learning has been used, e.g., decision tress, support vector machines, neural networks, ensemble methods (Iliou et al. 2019; Lagopoulos and Tsoumakas 2020; Lysenko et al. 2020; Rahman and Tomar 2021; Rovetta et al. 2017; Ustebay et al. 2019). Unsupervised learning has also been investigated (Alam et al. 2014; Suchacka and Iwański 2020; Rovetta et al. 2020; Zabihi et al. 2014). Although offline methods allows one to learn about features of bot traffic and lay the groundwork for novel online methods, this kind of approach is unable to recognize active bots. Here the online bot detection comes into play, by observing incoming requests within active sessions and trying to infer a user type as early as possible (Doran and Gokhale 2016; Suchacka et al. 2021). Regarding time series analysis, some previous works applied supervised classification techniques for game bots (Bernardi et al. 2017), network bots (Bonneton et al. 2015), and Web bots (Chen and Feng 2013).

Research questions under consideration in this paper are stated as follows. Is it possible to separate bots from humans on a Web server by analyzing only an initial part of a stream of Web client's requests for some (possibly short) period of session duration? What similarity measures are most adequate for time series clustering in the considered scenario?

To deal with the aforementioned issues, we represent each Web session on a server as a time series whose consecutive elements correspond to numbers of requests received from a given client in subsequent one-second intervals. We consider only beginnings of the series within a certain period of session duration and apply time series clustering to distinguish between two classes of clients: *1* (bots) and *0* (humans). Two well-known clustering techniques are implemented: hierarchical clustering and partition-based clustering, each with 23 different similarity measures. This approach is applied to real data obtained from Web server access logs.

The reminder of the paper is organized as follows. The next section presents preliminaries on time series clustering and similarity measures. Then the research methodology is presented, followed by a discussion of experimental results. The last section concludes the paper and outlines prospective directions of future work.

PRELIMINARIES

Time Series Clustering

Let us consider a dataset of n time series data $D = \{F_1, F_2, \dots, F_n\}$, where F_l is a sequence of values measured in time, $l = 1, 2, \dots, n$. *Time series clustering* is the process of unsupervised partitioning of D into a set of clusters $C = \{C_1, C_2, \dots, C_k\}$ in such a way that similar time series are grouped together based on a certain similarity measure, $D = \cup_{i=1}^{k} C_i$ and the clusters are disjoint: $C_i \cap C_j = \emptyset$ for $i \neq j$, $i = 1, 2, \dots, k$ and $j = 1, 2, \dots, k$ (Aghabozorgi et al. 2015).

Communications of the ECMS, Volume 36, Issue 1,
Proceedings, ©ECMS Ibrahim A. Hameed, Agus Hasan,
Saleh Abdel-Afou Alaliyat (Editors) 2022
ISBN: 978-3-937436-77-7/978-3-937436-76-0(CD) ISSN 2522-2414

We consider univariate time series, which are sequences of real numbers collected regularly in time, where each number represents a value – the number of requests received from a given Web client in one-second interval. Furthermore, all pre-processed time series subject to clustering have exactly the same length.

Time Series Clustering Methods

Methods for clustering time series may be broadly classified into six groups: hierarchical, partitioning, model-based, grid-based, density-based, and multi-step clustering (Aghabozorgi et al. 2015; Kotsakos et al. 2018). We apply two algorithms for the most popular clustering types: hierarchical and partitioning clustering.

Time Series Distance Measures

Time series clustering is not easy due to such common characteristics of time series as the presence of noise, outliers, and shifts in data. A key issue is finding similar time series is applying the appropriate way of calculating the similarity of data sequences, i.e., an adoption of an adequate similarity measure (distance measure). The concept of similarity in the context of time series is complex due to the dynamic character and high dimensionality of time dependent data.

In practice, calculation of the distance between time series is often approximated with the use of various methods. There is a wide choice of similarity measures for time series clustering (Aghabozorgi et al. 2015; Montero and Vilar 2015). The most common are distance measures of three types.

- *Model-free measures*. These basic measures are suitable especially for time series with the equal length. They often operate by comparing original time series or sequences of serial features extracted from them, like correlations, autocorrelations, spectral features, or wavelet coefficients.
- *Model-based measures*. The idea of these methods consists in fitting an underlying model to each time series and calculating the similarity between the fitted models. The most commonly used models are ARIMA and ARMA.
- *Comlexity-based measures*. Here the concept is to compare levels of complexity of time series by measuring the level of shared information by the compared series. The mutual information is approximated, usually using the notion of algorithmic entropy or Kolmogorov complexity.

We consider 23 different distance measures from all the three groups.

RESEARCH METHODOLOGY

This section discusses the proposed methodology to Web session time series clustering which involves the following steps:

- reading and pre-processing data from Web server logs to reconstruct user sessions and determine session features;
- assigning ground truth labels to sessions;
- transforming sessions to time series of a specified length;
- performing time series clustering with the use of various similarity measures;
- validating and comparing clustering results.

Building Time Series from Server Log Data

Extracting Sessions and Session Features

Raw data used in the study are entries recorded in a standard Web server access log of an e-commerce Web server. Each text line in a log file corresponds to a single HTTP request and contains information on the Web client's IP address, user-agent string, identifier of the requested server resource, timestamp, and other.

After pre-processing of request data Web sessions may be reconstructed. A *Web session* is defined as a sequence of HTTP requests (with more than one request) received from a Web client during a single visit to a website hosted by the server. A Web client is identified with an IP address and a user-agent string. An additional assumption is a minimum time gap between any consecutive sessions of a given client, equal to 30 minutes.

From individual fields of requests making up a session some session features may be determined, like the session duration (time interval in seconds), the session length (the total number of requests in session), the mean inter-request time, etc. Session features are useful for building a session representation for data mining approaches, as well as to identify some part of Web clients as robots based on heuristic rules.

Assigning Ground Truth Labels

The next step is session labeling with ground-truth labels. Clustering of observations from a given dataset is a task of unsupervised learning so the information about class labels is not used while generating clusters. However, this information is useful for other purposes, e.g., to perform the exploratory data analysis, to create an appropriate composition of an ultimate dataset used in experiments, or to evaluate the clustering results with the use of external indexes. Thus, the process of labeling Web sessions as bots (class *1*) or humans (class *0*) should be carried out with the greatest care and with the application of as many criteria as possible.

Our labeling procedure was described in (Suchacka and Motyka 2018) in detail. It is based on two databases of IP addresses and user-agents known to correspond to different kinds of Web browsers and bots: *Udger* database (Udger 2021) and *User agents* database (User-agents 2014). A big advantage of using *Udger* is the fact that most sessions labeled in this way have not only the class assigned (*0* or *1*) but also the Web client category, name, and version. Sessions performed by clients identified by *Udger* as bots without a client category or name were assigned to category "*Uncategorized Udger bot*".

Some of sessions that could not be labeled with *Udger* data were flagged as bots by using additional criteria: by performing a syntactic analysis of user-agents for bot-

related keywords (category "*Unknown bot – keyword*"), observing a request for *robots.txt* file in session, or applying heuristic rules for session features. These rules included: zero image-to-page ratio, all requests with empty referrers, all responses erroneous (with 4xx HTTP codes), and all requests of type HEAD. These sessions were assigned to category "*Unknown bot – heuristics*".
Session reconstruction and labeling was accomplished using our log analyzer implemented in C#.

Representing Sessions as Trimmed Time Series
Based on request timestamps sessions were transformed into time series, each element of which denotes how many requests arrived in a given second of the session duration. The next step was to decide what sequence length should be chosen for the time series analysis.
Web sessions naturally differ in terms of duration. Some bots, for instance, tend to request only one or two requests within an extended period of time whereas other may perform extremely long-lasting sessions. On the other hand, human users may access only one page of the website if they are not interested in the contents or may spend up to several dozen minutes browsing the online store offer and selecting items they are willing to buy.
Having in mind the necessity of recognizing bot sessions as soon as possible, we aimed at considering relatively short session duration. Moreover, the shorter the sequence is, the more sessions from the original dataset are left in the ultimate dataset. Finally, the time series length equal to 100 seconds was chosen. All the time series corresponding to sessions lasting at least 100 seconds were trimmed to 100 elements whereas all shorter time series were excluded from the analysis. We excluded shorter sequences in this study because our goal was to apply many time series similarity measures, some of which are limited to time series with equal length.
A program to analyze and transform the session dataset into the time series dataset was implemented in R.

Distance Measures Used

Time series clustering was implemented in R with the use of package *TSclust* (Montero and Vilar 2020). We applied 23 similarity measures available in the package, which include a wide range of approaches to calculate the distance between time series. The measures and their shortcut names used hereafter are as follows.

1. *Model-free distance measures*

1.1. Simple measures:
- EUCL – Euclidean distance,
- FRECHET – Fréchet distance,
- DTW – Dynamic Time Warping distance,
- CORT – distance based on the first order temporal correlation coefficient.

1.2. Measures based on correlations and autocorrelations:
- COR1 – correlation-based distance,
- COR2 – correlation-based distance with the parameter allowing regulation of the fast distance decreasing,
- ACF – autocorrelation-based distance,
- PACF – distance using the partial autocorrelation function.

1.3. Measures based on periodograms:
- PER – Euclidean distance between the periodogram ordinates,
- PER_NP – Euclidean distance between the normalized periodogram ordinates,
- PER_LNP – Euclidean distance based on the logarithm of the normalized periodogram,
- PER_INT – distance based on the integrated periodograms (cumulative versions of the periodograms).

1.4. Measures based on nonparametric spectral estimators:
- SPEC_LLR_LS – distance with the spectra replaced by the exponential transformation of local linear smoothers of the log-periodograms, obtained via least squares,
- SPEC_LLR_LK – distance with the spectra estimated by the exponential transformation of local linear smoothers of the log-periodograms, obtained by using the maximum local likelihood criterion,
- SPEC_GLK – distance using the local maximum log-likelihood estimator computed by local linear fitting,
- SPEC_ISD – distance evaluating the integrated squared differences between nonparametric estimators of the log-spectra using local linear smoothers of the log-periodograms, obtained by using the maximum local likelihood criterion.

2. *Model-based distance measures*
- AR_PIC – Piccolo distance – Euclidean distance between autoregressive approximations of ARIMA structures,
- AR_MAH – Maharaj distance, based on autoregressive approximations of ARMA structures,
- AR_LPC_CEPS – cepstral-based distance using linear predictive coding (LPC) cepstrum for ARIMA time series.

3. *Complexity-based distance measures*
- CID – Euclidean distance with a complexity correction factor,
- PDC – distance based on divergence between permutation distributions of order patterns in m-embedding of the original series,
- CDM – compression-based distance measure,
- NCD – a simplified version of the compression-based distance measure.

Clustering Algorithms Used

Time series representing Web sessions were clustered with the use of two well-known algorithms.
The first algorithm is agglomerative hierarchical clustering with complete linkage cluster selection. The algorithm generates a hierarchy of clusters starting from each time series being a separate cluster. Then, it gradually merges the most similar pairs of clusters until

the desired number of clusters is obtained. A complete linkage cluster selection means that to decide which two clusters are closest to each other in each step, the distance between any two clusters is defined as the longest distance among all their member time series.
The second algorithm is Partitioning Around Medoids (PAM), also known as k-Medoids. The algorithm partitions all the time series into k groups so that each group contains at least one series. Each cluster has a medoid prototype, which is the most centrally located object in the cluster (i.e., the time series whose average distance to all other series in the cluster is minimum).
The hierarchical clustering was conducted with the use of *hclust* function available in package "stats" and the partitioning clustering was done with *pam* function from package "cluster". In both cases the target number of clusters was two.

Clustering and Performance Evaluation

Bootstrap Sampling
In reality, Web sessions observed in a given time window are not balanced in terms of the client classes and categories which may negatively affect the performance of machine learning algorithms. Furthermore, it is necessary to perform clustering in reasonable time whereas some algorithms for computing time series similarity matrix have high computational complexity. To prevent a possible bias of clustering results and provide reasonable computation time, the ultimate dataset of time series used in experiments was created via the bootstrap sampling. This under-sampling method consists in drawing sample observations repeatedly with replacement from the source dataset to be used in a single experiment run. The ultimate dataset was created by drawing proportionate numbers of time series of different client categories (see subsection "Bootstrap Datasets"). The time series clustering with the use of 23 distance measures and two clustering algorithms was repeated ten times for different bootstrap subsets. The final results are averaged performance scores of all experiment runs.

Clustering Performance Measure
To evaluate the clustering quality we applied an external index $Sim(C, C')$ (Gavrilov et al. 2000; Liao 2005; Montero and Vilar 2020). Given the ground-truth cluster partition $C = \{C_1, \dots, C_k\}$ and the experimental partition $C' = \{C'_1, \dots, C'_k\}$, the similarity index expresses the agreement between these two solutions:

$$Sim(C, C') = \frac{1}{k}\sum_{i=1}^{k} \max_{1 \le j \le k} Sim(C_i, C'_j),$$

where

$$Sim(C_i, C'_j) = \frac{2|C_i \cap C'_j|}{|C_i| + |C'_j|},$$

where $|\cdot|$ denotes the cardinality of the elements in the set. The higher score is achieved, the better clustering results are. The score of 1 means that the two clusterings are the same and the value 0 means they are completely dissimilar.

RESULTS AND DISCUSSION

Data Description

Data used in the experiment were obtained from 20-hour log file for an e-commerce website, recorded in November 2019. As a result of data pre-processing, session reconstruction, and excluding single requests and admin sessions, there were 2,218 sessions in total, including 1,371 humans and 847 (38.19%) bots. The session length varied from 2 to 5,270 requests. The longest session lasted for 84,849 seconds, i.e., almost 24 hours (it was clearly a bot session).
Fig. 1 and Fig. 2 plot time series of humans and bots, respectively, taking into consideration the first five minutes of session. Traffic patterns of both groups clearly differ. Since humans navigate through the website via Web browsers, their traffic patterns reflect the way of downloading server resources by browsers: one can observe clear peaks after requesting subsequent pages with the corresponding embedded objects, separated by the users' think time, required for browsing and analyzing the page contents (Fig. 1). On the other hand, artificial agents parse the website according to the implemented algorithms – their requests are less frequent and more regular in nature (Fig. 2).

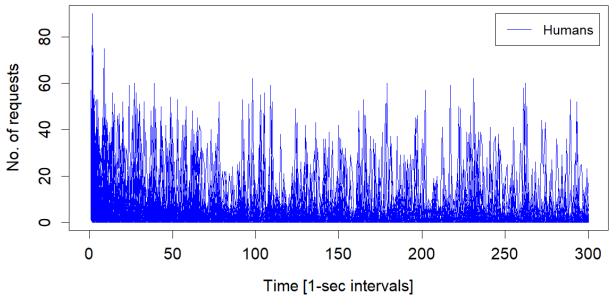


Figure 1: Visualization of Human Users' Traffic During the First 300 Seconds of Sessions' Duration

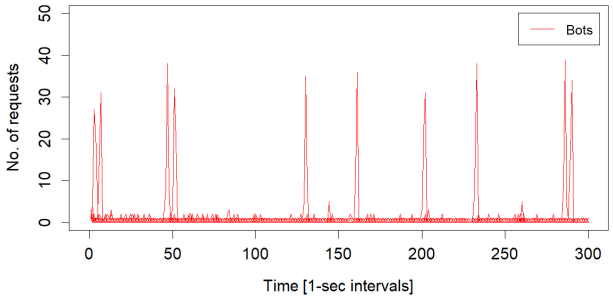


Figure 2: Visualization of Bots' Traffic During the First 300 Seconds of Sessions' Duration

Bootstrap Datasets

After transforming sessions to 100-second time series (and eliminating shorter sessions), the ultimate dataset to be analyzed contained 980 time series: 543 class *0* series

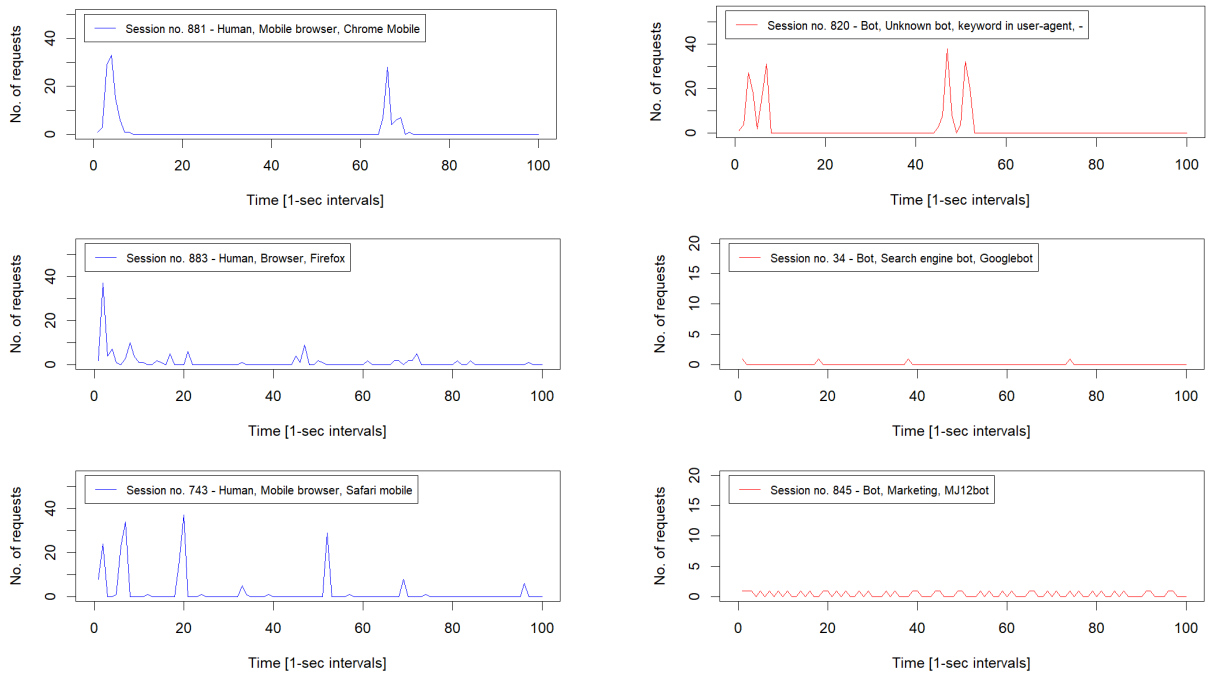


Figure 3: Visualization of Example Time Series of Class *0* (**left**) and of Class *1* (**right**)

(human sessions) and 437 class *1* series (bot sessions). Table 1 presents how many time series were from individual classes and client categories. Numbers of sessions completed via mobile and desktop browsers were roughly equal (276 and 267 time series, respectively). Regarding robot sessions, the most numerous were unknown bots identified based on heuristic rules (165 series), marketing bots (133 series), and search engine bots (120 series).

Table 1: Distribution of 100-second Time Series According to Classes and Categories

Client class	Client category	No. of sessions
0	*Browser*	267
0	*Mobile browser*	276
1	*Marketing*	133
1	*Search engine bot*	120
1	*Uncategorized Udger bot*	18
1	*Unknown bot – heuristics*	165
1	*Unknown bot – keyword*	1

Fig. 3 visualizes several samples of time series corresponding to sessions from both classes. It can be seen in Fig. 3-left that sessions generated by human-operated Web browsers reveal similar patterns regardless of a browser name: spikes in the number of requests correspond to successive user clicks (page views involve requests for page description files and embedded objects, like images, pdf documents, etc.). Plots in Fig. 3-right show that the Web traffic generated by bots may be more differentiated depending on a specific crawling software. Sessions no. 34 and 845 show a very different traffic characteristics than the human sessions (no. 881, 883, 743) – these are benign bots which reveal their identities (Googlebot, MJ12bot) in user-agent fields. In contrast, session no. 820 is a bot emulating human behavior.

Each bootstrap dataset had the following composition regarding time series from different client categories:

- Browser: 20,
- Mobile browser: 20,
- Marketing: 10,
- Search engine bot: 10,
- Uncategorized Udger bot: 10,
- Unknown bot – heuristics: 10,
- Unknown bot – keyword: 1.

Clustering Results

Fig. 4 shows clustering performance scores obtained for various distance measures and clustering algorithms as a degree of agreement between the ground truth-based partition of time series and the experimental solutions. One can observe that the clustering quality highly depends both on a distance measure and a clustering method applied. In general, much better results were achieved with the partition-based method than the hierarchical one – results were higher for nearly all similarity measures applied.

Regarding the similarity measures, the lowest scores were achieved for the periodogram-based measures (PER, PER_NP, PER_LNP, PER_INT) and for the model-based measures (AR_MAH, AR_LPC_CEPS, AR_PIC). Scores of the simple measures (EUCL, FRECHET, DTW, CORT), as well as of the measures based on correlations and autocorrelations (COR1, COR2, ACF, PACF) are pretty similar (relatively low for the hierarchical clustering algorithm and moderate for PAM).

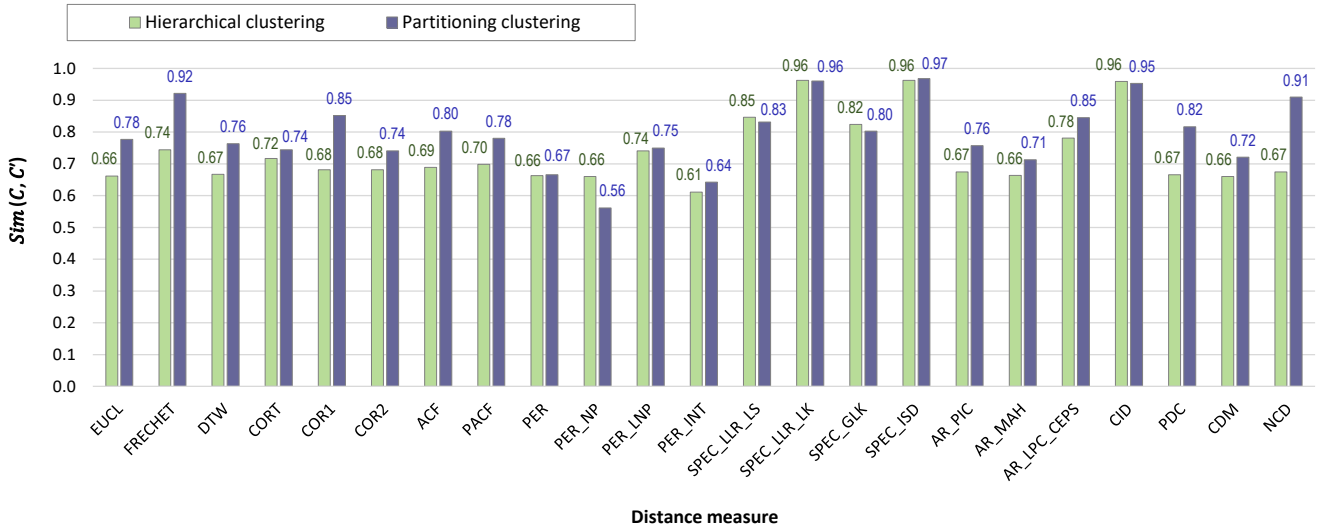


Figure 4: Clustering Performance Depending on a Distance Measure and a Clustering Method

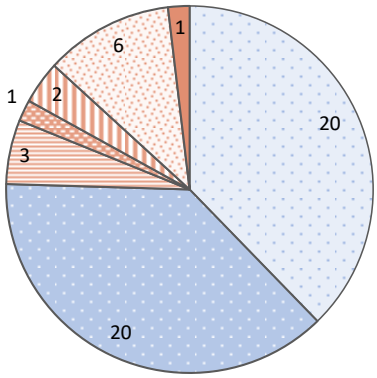
Among these solutions only the result achieved for PAM with the Fréchet distance (FRECHET) stands out positively – it should be emphasized, however, that calculating the time series similarity matrix with the Fréchet method takes the longest time, which may disqualify this method for use in real-time applications, such as the online bot detection. The efficiency of the complexity-based distance measures is difficult to generalize – in this type of approaches CID measure clearly stands out positively for both clustering algorithms.

A very effective approach to assessing the time series similarity in the scenario under consideration turned out to be the use of nonparametric spectral estimators (SPEC_LLR_LS, SPEC_LLR_LK, SPEC_GLK, SPEC_ISD measures). These methods, along with the Euclidean distance with a complexity correction factor (CID), achieved the highest average efficiency as regards the *Sim* index.

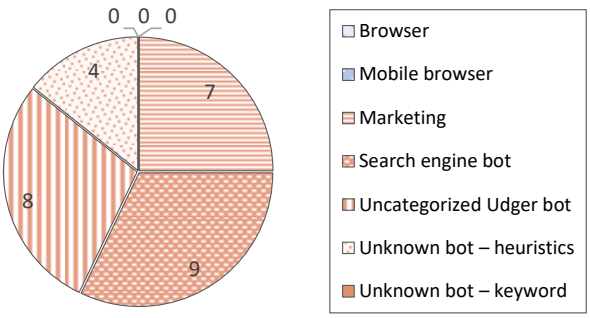
To sum up, the best clustering results of all the distance measures in the scenario under consideraion were clearly achieved by three measures: SPEC_LLR_LK and SPEC_ISD, based on nonparametric spectral estimators, and CID, the Euclidean distance with a complexity correction factor. These measures not only provided the highest efficiency of clustering time series representing bot and human sessions (performance scores of at least 0.95) but they were also insensitive to a clustering method (hierarchical or partitioning).

Let us investigate results for the best case in more detail, i.e., for SPEC_ISD distance measure and the partitioning clustering. Fig. 5 visualizes the composition of generated clusters in terms of the number of sessions from different client categories (and thereby, from different classes; class 0 is marked in shades of red, class 1 – in shades of blue). Cluster 1 is much larger – it contains 53 time series, 40 of which are of class *0* (20 Browsers and 20 Mobile browsers) and only 13 series of class *1* (from all five bot categories). Cluster 2 contains only 28 time series but all of them belong to class *1*.

One can see that in this case of time series clustering it was possible to separate bots from humans up to a certain degree but not completely. This confirms that robots' online behaviors are highly differentiated, as opposed to navigational patterns demonstrated by human users, which were all gathered in one cluster. This observation clearly demonstrates that a larger number of clusters should be considered.



(a) Cluster 1 (Humans Mostly)



(b) Cluster 2 (Bots Only)

Figure 5: Composition of Clusters for the Best Case (SPEC_ISD Distance Measure, Partitioning Clustering)

CONCLUSIONS

The study discussed in this paper showed that the proposed way of clustering Web sessions of bots and humans, represented as time series of limited length, may give very good results provided that an appropriate distance measure is used. In order to draw a sound conclusion regarding the best measure, however, a larger set of experiments should be carried out because it is not sure if the same measure will give accurate answers in different scenarios.

Our prospective works include investigating time series clustering for Web session scenario taking into consideration other session features, like the number of page or image requests, the amount of data transferred to the client, as well as various lengths of time series being clustered. Other possible research direction is to increase the number of generated clusters with respect to Web client categories (i.e., clustering of multi-label time series). Furthermore, it would be undoubtedly worth performing experiments for a bigger dataset of Web sessions to embrace less common bots, like Web scrapers, attacking bots, fake crawlers, etc.

We are also planning to develop an effective online bot detection method based on time series clustering. The experimental results discussed in this paper show a big potential of the proposed approach to develop a method for identifying Web bots on the fly. By observing the initial progress of active Web sessions and comparing their request arrival patterns with prototypes of previously generated clusters, an early decision on classifying the client as bot or human might be determined.

REFERENCES

Aghabozorgi, S.; A.S. Shirkhorshidi; and T.Y. Wah. 2015. "Time-series clustering – a decade review." *Information Systems* 53, 16-38.

Alam, S.; G. Dobbie; Y.S. Koh; and P. Riddle. 2014. "Web bots detection using Particle Swarm Optimization based clustering". In *Proc. IEEE CEC'14*, 2955-2962.

"Bad Bot Report 2021: The Pandemic of the Internet". 2021. Technical Report, Imperva Incapsula, https://www.imperva.com/resources/resource-library/reports/bad-bot-report/.

Bernardi, M.L.; M. Cimitile; F. Martinelli; and F. Mercaldo. 2017. "A time series classification approach to game bot detection". In *Proc. of WIMS'17*, Article 6.

Bonneton, A.; D. Migault; S. Senecal; and N. Kheir. 2015. "DGA bot detection with time series decision trees". In *Proc. of BADGERS'15*, 42-53.

Chen, Z. and W. Feng. 2013. "Detecting impolite crawler by using time series analysis". In *Proc. of ICTAI'13*, 123-126.

Doran, D. and S.S. Gokhale. 2016. "An integrated method for real time and offline web robot detection," *Expert Syst.* 33 (6) 592-606.

Gavrilov, M; D. Anguelov; P. Indyk; and R. Motwani. 2000. "Mining the stock market: which measure is best? (extended abstract) ". In *Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 487-496.

Iliou C.; T. Kostoulas; T. Tsikrika; V. Katos; S. Vrochidis; and Y. Kompatsiaris. 2019. "Towards a framework for detecting advanced Web bots". In *Proc. of ARES'19*, Article no. 18.

Kotsakos, D.; G. Trajcevski; D. Gunopulos; and C.C. Aggarwal. 2018. "Time-series data clustering". In *Data Clustering*. Chapman and Hall/CRC, pp. 357-380.

Lagopoulos, A. and G. Tsoumakas. 2020. "Content-aware Web robot detection." *Applied Intelligence* 50(11), 4017-4028.

Liao, T.W. 2005. "Clustering of time series data – a survey." *Pattern Recognition* 38(11), 1857-1874.

Lysenko, S.; K. Bobrovnikova; P.T. Popov; V. Kharchenko; and D. Medzatyi. 2020. "Spyware detection technique based on reinforcement learning". In *Proc. of CEUR Workshop*, vol. 2623, 307-316.

Montero, P. and J.A. Vilar. 2015. "TSclust: An R package for time series clustering." *J. Stat. Softw.* 62(1), 1-43.

Montero, P. and J.A. Vilar. 2020. "Package TSclust". https://cran.r-project.org/web/packages/TSclust/TSclust.pdf.

Rahman, R.U. and D.S. Tomar. 2021. "Threats of price scraping on e-commerce websites: attack model and its detection using neural network." *Journal of Computer Virology and Hacking Techniques* 17(1), 75-89.

Rovetta, S.; A. Cabri; F. Masulli; and G. Suchacka. 2017. "Bot or not? A case study on bot recognition from Web session logs". In *Quantifying and Processing Biomedical and Behavioral Signals*, SIST 103. Springer, 197-206.

Rovetta, S.; G. Suchacka; and F. Masulli. 2020. "Bot recognition in a Web store: An approach based on unsupervised learning." *J. Netw. Comput. Appl.* 157, 102577.

Suchacka, G.; A. Cabri; S. Rovetta; and F. Masulli. 2021. "Efficient on-the-fly Web bot detection." *Know.-Based Syst.* 223, 107074.

Suchacka, G. and J. Iwański. 2020. "Identifying legitimate Web users and bots with different traffic profiles – an Information Bottleneck approach." *Know.-Based Syst.* 197, 105875.

Suchacka, G. and I. Motyka. 2018. "Efficiency analysis of resource request patterns in classification of Web robots and humans". In *Proc. of ECMS'18*, 475-481.

Udger. 2021. https://udger.com (access: July 12, 2021).

User-agents. 2014. http://www.user-agents.org (access: September 4, 2017).

Ustebay, S.; Z. Turgut; and M.A. Aydin. 2019. "Cyber attack detection by using neural network approaches: shallow neural network, deep neural network and autoencoder". In *Proc. of CN'19*, 144-155.

Zabihi, M.; M.V. Jahan; and J. Hamidzadeh. 2014. "A density based clustering approach to distinguish between Web robot and human requests to a Web server." *ISC Int. J. Inf. Secur.* 6 (1), 77-89.

AUTHOR BIOGRAPHY

GRAŻYNA SUCHACKA received the M.Sc. degrees in Computer Science and in Management, as well as the Ph.D. degree in Computer Science with distinction from Wroclaw University of Science and Technology, Poland. Now she is an Assistant Professor in the Institute of Informatics at the University of Opole, Poland. Her research interests include data analysis and modeling, data mining, machine learning, and Quality of Web Service with special regard to bot detection and electronic commerce support. Her e-mail address is: gsuchacka@uni.opole.pl.