Web Bot Detection Based on Hidden Features of HTTP Access Log

Kaiyuan Li[1], Mingrong Xiang[2(✉)], Mitalkumar Kakaiya[1], Shashank Kaul[1], and Xiaodong Wang[3]

[1] Webjet Limited, Melbourne, Australia
[2] Deakin University, Geelong, Australia
mxiang@deakin.edu.au
[3] Victoria University, Melbourne, Australia
tony.wang@vu.edu.au
https://www.webjetlimited.com, https://www.deakin.edu.au, https://www.vu.edu.au/

Abstract. Web bot generates a large fraction of traffic on present-day Web servers. It not only introduces a threat to website security, performance and user privacy but also raises concerns about valuable information and digital asset scripting. Much research explored traffic features, tagging legitimate users and bot traffic, and created some efficient machine-learning models to detect web bots. However, previous machine learning methods used to detect web bots based on the observable raw data, that have become more challenging with the increasingly diverse and complex logic and technologies of web bots. In this research, we proposed the Autoencoder-based method to detect the web bot, distinguishing the HTTP access behaviours between humans and web bots. Our method aims to find the hidden features from the raw HTTP access data and allow for clustering the web bots with scattered raw features. Furthermore, we use the polar coordinates transformation strategy to rotate the geometry of hidden features and solve the clustering difficulties caused by the randomness of the neural network environment. We compare the web bot detection performance with the other competitors, which yielded about 30% improvements in accuracy.

Keywords: Web bot · Web bot detection · Unsupervised learning · Machine learning · Cyber security

1 Introduction

In the present era, the Internet is a core part of our everyday life, while various online activities enrich daily activities. A web bot is a software tool that carries out specific Web tasks, also called an Internet robot or artificial agent. The web bot is usually autonomous, following the structure of hyperlinks according to a specific algorithm [1,2]. Although many bots are positive and valuable (e.g. search engine

S. Yu et al. (Eds.): TridentCom 2022, LNICST 489, pp. 32–43, 2023.
https://doi.org/10.1007/978-3-031-33458-0_3

crawlers) [3], bad bot scripting or malicious activities against those Internet service providers to make profits and damage end-user privacy [4–6].

According to recent bot traffic reports [7], bot traffic made up 42.3% of all internet activities in 2021, up from 40.8% in 2020. Bad bot traffic accounted for almost 28% of global web traffic in 2021, nearly double that of the so-called "good bot". Those bad bots are now more advanced and evasive than ever, mimicking human behaviours in ways that make them harder to be detected and prevented [8,9].

Along with the increasing proliferation and sophistication of bad web bot, the losses suffered by web companies have driven much more research on robot traffic analysis and detection, so protection tools like CAPTCHA were invented to help detect bot in a hard way [10,11]. In recent research [12], the researchers presented a way to extract the features of the HTTP access log of an online bookstore and provided a comparison of bot detection performance among some well-known machine learning methods. Suchacka, G et al. [13] proposed a decision tree-based neural network method for effective on-the-fly Web bot detection.

However, those presented works focus on observable data features and do not consider hidden feature representations. With the development of technology, the behaviours of bad web bots are becoming more and more complex displaying different acted patterns when they hack webs. Therefore, it is necessary to find bot-related hidden features in the data and accurately distinguish them from human behaviours.

In this research, we have extracted the data from the HTTP access log of the online ticketing agent and proposed an unsupervised method for bot detection. Across all industry sectors, online ticketing agents as part of the e-commerce category ranked fifth in terms of bad bot traffic intensity and ranked No.1 in terms of complex bot traffic [14]. Our method is based on the autoencoder [15] and is used to find the hidden features of the web access activities for humans and bots and provide high-accuracy bot detection results. Our main contributions are:

- We demonstrated that directly applying such well-known unsupervised methods (e.g. K-means, C-mean, MeanShift and Agglomerative Clustering) does not work for bot detection based on extracting information from HTTP access log raw data from an online ticketing agent.
- We introduced the feature extraction and preprocessing approaches for the HTTP access log, which can also be applied to the other data resources (different web systems).
- We proposed an autoencoder-based method to extract the hidden feature of web access patterns. The bot detection performance is greatly improved when using the hidden features compared to directly doing the clustering on data.
- We proposed to use the polar coordinates strategy to solve the clustering difficulties caused by the environmental randomness of the neural network.

2 Related Work

Many successful methods have been proposed to classify the bot and human HTTP access behaviours and show satisfying performance. DeepDefense [16] used the RNN (recurrent neural network) to capture the features of the HTTP access log segments, which did the classification based on the log context information. Cabri A et al. [17] proposed a binary classification method of multivariate data streams from web servers to identify ongoing user sessions generated by bots or humans. BotGraph [18] utilized the sitemap with CNN to detect the inner behaviour of bots, showing high performance. Unfortunately, these methods require the labelled training set, which is costly and difficult to get in real applications.

Rovetta S et al. [12] proposed to use an unsupervised C-mean clustering method for the bot detection and showed a higher performance than K-means, and even better than supervised MLP and SVM. It is a promising direction that mitigates the impact of the need for labelled data in real applications. However, those presented works focus on the observable data, which has become continuously difficult to detect the more complex and intelligent web bot behaviours. The web bot has different HTTP access patterns or features [19,20] that challenge people to find similar hidden features of different web bot accesses to improve the bot detection performance.

In contrast, our method can find the hidden features of different web bots and can be easily applied with the clustering method to do the classification between web bots and humans. Our proposed method is based on the autoencoder that includes a neural network constructed decoder and encoder blocks. We used the encoder results as the hidden features of the HTTP access log and showed a higher performance compared to directly applying the clustering methods to the original dataset.

3 Problem Statement

In order to do the clustering of HTTP access activities between the human and web bot, we expect them to exhibit two distinct cluster distributions and geometries. However, web bots were generally built under different logic and technologies and presented with different feature embedding. Hence, we will face several problems as follows.

3.1 Complex and Scattered Distribution Patterns of Web Bot

The different web bots exhibit different web access behaviours with various hacking purposes. Therefore, the data we extracted from the HTTP Access logs are in various complex and scattered distributions. As shown in Fig. 1, we have visualized the raw data distribution by using UMAP [21]. The samples of human are labelled as 0, and the samples of web bot are labelled as 1. We can find those data distributions of web bots are scattered and do not have flat geometry

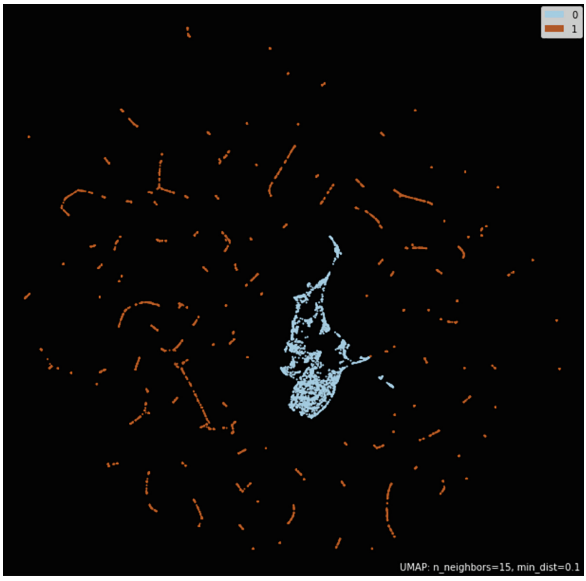


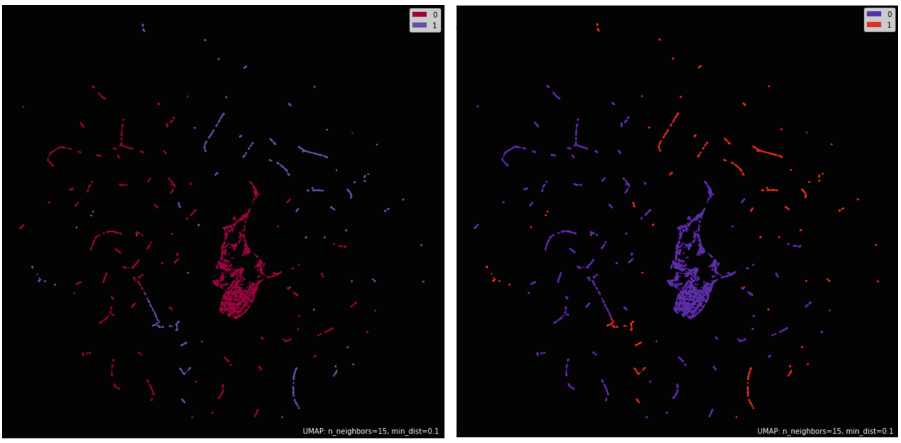
Fig. 1. Shows the raw data distribution, which are the features of inbound HTTP requests. The blue points indicate the human samples that are labelled as 0. The brown points are the web bot labelled as 1. The human distributions have a flat geometric pattern, but the web bots show the various and complex distributions (Color figure online)

characteristics. Hence, it is impossible to cluster the human and web bot directly based on distance (e,g. Euclidean distance).

Hierarchical clustering-based methods are proposed to solve the clustering problem with none Euclidean distances [22]. For example, Agglomerative Clustering performed bottom-up Hierarchical clustering strategy [23], i.e., each observation starts with its own cluster and the clusters are successively merged together. However, we found that either directly using the K-means (Euclidean-based) or the Agglomerative Clustering (none Euclidean) on the raw data of the HTTP access logs is not working.

As shown in Fig. 2, the sub-Figure 2(a) and 2(b) show the label prediction results after clustering. In the (a) and (b) the predicted label 0 indicates the samples of humans, and the predicted label 1 presents the web bot. Compared with the labelled raw data in Fig. 1, most of the web bots are predicted as human, which is incorrect. Eighter Euclidean-based (K-means) or none Euclidean (Agglomerative Clustering) cannot distinguish most of the HTTP access activities of web bots from humans. Therefore, it is important to find the hidden feature representations for all samples of web bots, which allows the feature embedding of web bots to be clustered together.

In this research, we proposed an Autoencoder-based method to find the hidden features of HTTP access data, which significantly cluster all the samples of web bots together and clearly distinguished them from humans.



(a) K-means (b) Agglomerative Clustering

Fig. 2. Shows the human and web bot prediction results of K-means (sub-figure (a)) and Agglomerative Clustering (sub-figure (b)). The results show that neither Euclidean (K-means) nor non-Euclidean (Agglomerative Clustering) is sufficient to classify the human and web bot based on raw data. Lots of the web bot samples are mispredicted as human.

3.2 Difficulties of Clustering on Encoder Results

Another problem is that when we map features into a low-dimensional space, the distribution of features is affected by the environmental randomness of the neural network. Hence, It will cause angular uncertainty in the geometric shapes representing web bots and humans, so distances sometimes do not discriminate effectively in low-dimensional spatial representations. As shown in Fig. 3, sub-figure (a) and (b) are hidden features embedding two experiments of our method. The red points indicate the human samples, and the blue points are the web bot samples. As the hidden features show in the sub-figure (a), we can directly use the K-means to cluster the data, but in sub-figure (b), the bottom left area of the blue points will miss predicted bot as the human (the detailed experimental results are presented in Sect. 5).

In this case, we use a simple rotation method to dominate the hidden features that can sufficiently work with K-means clustering. As presented in Sect. 4.2, we convert the low-dimensional hidden feature maps to polar coordinates, which successfully allow the K-means to cluster the web bot and human.

4 Data Extraction and Method

4.1 Data Extraction

Source data are HTTP access logs across Webjet's large numbers of web servers, in terms of a range of customers' activities including flight searches, itinerary

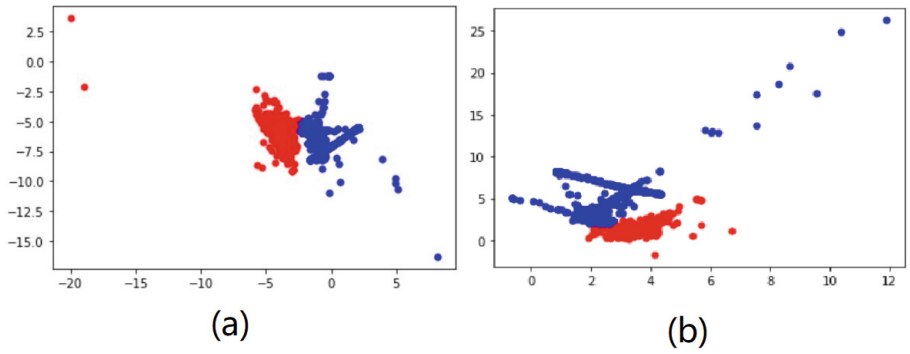


Fig. 3. The Encoder block generates two examples of hidden feature embedding. The red points indicate the human, and the blue points are web bots. In sub-figure (a), the embedding of human and web bot got the flat geometric and easy to be clustered by K-means. In sub-figure (b), the angle of geometric patterns causes clustering difficulties in a low-dimensional space. (Color figure online)

details, add to the shopping cart, payment reviews, .etc. There are quite a lot of sub-modules integrated with each main function, such as airport autocomplete functions, price calendars and so on. HTTP access logs are collected in 2022 and further processed by Sumolgic comprehensive queries to get expected statistics data. Each row represents a whole day of activities from an individual client IP, the actual IP address replaced by the encrypted unique Hash value.

In statistics data, features are extracted into two categories, listed in Table 1, the first category indicates commonly used information in many previous bot detection studies [24–26]. For example, in the Table 1, vol, is the total volume of data sent from a client, that should be within a specific range. According to the logic of the web services, normal user requests would exactly follow this logic. Also, in general bot developers would like to put some additional common information to save time to determining which information is required case by case. And the DDoS bot would be happy to carry even more data to achieve the bandwidth attack goal. Also another example in Table 1, 4xx, represents 400–499 HTTP server response code [27,28]. Well-tested Web services should very little those codes at the user request. However, sometimes it will still happen. For example, invalid inputs. Another example, Clients probably saved the URL to somewhere like browser bookmarks, but if the web admin on the server side removed that URL endpoint, the user would see a 404 response code. Seeing too many 400–499 codes from a single client can be very suspicious.

The second category is the online ticket agent e-commerce-oriented features of Webjet.com.au, which represent core functional interaction between clients and web servers. For example, most bookings of tickets and hotels start with a search page, which means we should see API calls to perform searching functions, which are noFS, noHS and noPS in Table 1. Then, users usually go through the shopping cart review page to review and adjust items. Users should not go to

the payment page or booking confirmation page directly due to the nature of common sense and Webjet-designed Web services interaction flows.

Table 1. Detail of features extracted from the HTTP access log

Category	No.	Name	Type	Description
Common session features	1	req	int	Total number of requests in 24 h
	2	req	int	Total number of requests in 24 h
	3	vol	double	double Total volume of data sent from the client [KB]
	4	eRefR	[0,100]	Percentage of requests with empty referrer
	5	4xx	[0,100]	Percentage of erroneous requests response code 400–499
	6	total4xx	int	Number of 4xx types
	7	totalref	int	Number of referrers
Booking-oriented features	8	noH	int	Number of views of the website's home page
	9	noL	int	Number of login operations
	10	noFS	eRefR	Number of flight search
	11	noHS	int	Number of hotel search
	12	noPS	int	Number of package search
	13	noCR	int	Number of shopping cart page request
	14	noPY	int	Number of payment page
	15	noCF	int	Number of confirmation

4.2 Method

The Autoencoder consists of Encoder and Decoder blocks, which are the two neural networks. Given a set of unlabelled HTTP access dataset $X \in R^{m \times n}$. The Encoder block E leans a non-linear transformation, which mapping the input space X into a latent space with low-dimension.

$$z = E(x) = W_E^2 \sigma(W_E^1 x + b_E^1) + b_E^2 \tag{1}$$

Here, σ is a non-linear activation function. W and b indicate the trainable weights and bias for different layers of the neural network. The latent vector z contains the critical information of input x. As expected, the samples of x contain similar properties and should have similar latent vectors (hidden features). Therefore, we can use the Encoder block to extract similar hidden features from the scattered distribution of web bots, enabling these feature embedding of network robots to adjacent locations in a low-dimensional space.

Moreover, the Decoder block D learns the non-linear transformation from Z to X.

$$\hat{x} = D(z) = \delta(W_D^2 \sigma(W_D^1 z + b_D^1) + b_D^2) \tag{2}$$

Here, the δ is the activation function sigmoid, which returns values of hidden features into [0,1]. As we get the decoded output $\hat{x}$, we can define a loss function to approximate the $\hat{x}$ to the raw high-dimensional input space x.

$$\text{Loss} = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{3}$$

After we trained an Autoencoder model, we can directly use the final latent vectors from the Encoder block for clustering and predicting the corresponding labels. However, as we mentioned in Sect. 3.2, the rotation operation is required for solving clustering difficulties caused by the neural network randomness. The low-dimensional latent vector z is in two dimensions for each sample. The equation for transforming the latent variables to polar coordinates is as follows:

$$\mathrm{r} = \sqrt{(\mathrm{z}_0)^2 + (\mathrm{z}_1)^2} \tag{4}$$

$$\theta = \frac{180 \times \mathrm{ArcTan}(\frac{\mathrm{z}_1}{\mathrm{z}_0})}{\pi} \tag{5}$$

r is the radius, and θ is the angular coordinate. Hence, we can have a new hidden feature representations $\hat{z} = [\mathrm{r}, \theta]$.

5 Experimental and Evaluation

In this research, we used the HTTP access log dataset extracted from the Webjet (the detailed information please refer to Sect. 4.1). This dataset contains 8,357 samples (random selected relatively rich data from raw statistics dataeset), 3,322 human access and 5,035 web bot access (bad), and the vector length of each sample is equal to 15. For all experiments, we adapted the Adam optimizer and set the learning rate equal to 1e−3. The non-linear activation function σ is ReLU. The output dimension of the Encoder block is equal to 2, which is the same as the input dimension of the Decoder block. The hidden units of the Encoder block and Decoder block are set to 5.

Baseline Methods. We did 10 trials and reported the average prediction accuracy to compare with all the competitors.

- K-means: Euclidean distance based-method [29]. It is a well-known and commonly used clustering method, which is frequently used as a baseline unsupervised method in many research.
- C-mean: The Fuzzy clustering method, which is a distance-based clustering method. It is reported to have the highest performance in recent research for web bot detection on online web store data [30].
- MeanShift: It is also distance-based data, but is designed for the uneven cluster size and non-flat geometry [31].
- Agglomerative Clustering: It is one type of Hierarchical clustering method. Agglomerative Clustering performs clustering with non-Euclidean distances and possibly connectivity constraints [23].

First, we have evaluated the polar coordinates transformation performance on our dataset. As shown in Fig. 4, there are three sub-figures. The sub-figure (a) shows the hidden features embedded with the ground truth label. The red points are human samples, and the blue points are web bot samples. We can find that most web bots' hidden representations are adjacent to each other by embedding

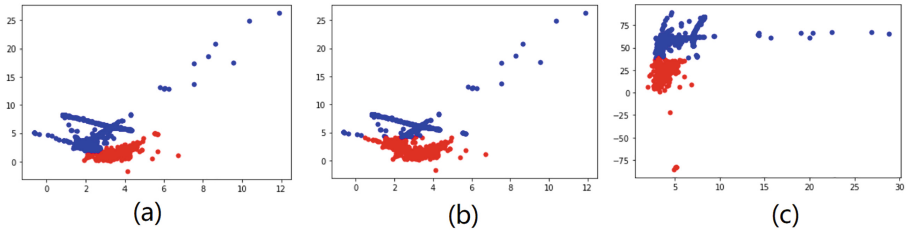


Fig. 4. An example of the hidden feature embedding with angular uncertainty is that web bots are mispredicted as humans. The sub-figure (a) shows the hidden feature embedding with the ground truth labels. The sub-figure (b) shows the embedding with the predicted labels that directly apply the K-means. The sub-figure (c) shows the K-means prediction results after performing the polar coordinates transformation, indicating that web bots are successfully distinguished from humans. (Color figure online)

the hidden features. As shown in sub-figure (b), if we directly apply K-means with the hidden features, then some of the web bot (blue) (on the bottom left) would be mispredicted as the human (red). In this case, we applied the polar coordinates transformation on the hidden features as shown in sub-figure (a), and got the new hidden feature representations for clustering. As shown in sub-figure (c), the embedding of the new hidden features were successfully clustered by the K-means.

After that, we evaluated the classification performance on the whole dataset, and compared it with different clustering methods and auto-loader with the polar coordinates method.

Table 2. Comparison of accuracy with other competitors

Method	Accuracy
K-means	0.652
C-mean	0.655
MeanShift	0.600
Agglomerative Clustering	0.650
Autoencoder	**0.900**
Autoencoder with polar coordinates	**0.995**

As shown in Table 2, web bots and human classification accuracy are similar when directly using the different clustering methods on raw data. The average accuracy of K-means is about 65.2%, while the MeanShift only has 60% accuracy. Although the MeanShift can handle problems of uneven cluster size and non-plat geometry, it had an unsatisfactory result on our dataset. We also did experiments by applying C-mean, which got the highest performance in the recent research

about bot detection of a web store [12]. The accuracy of C-mean is similar to K-means, which is about 65.5% in this case. While Agglomerative Clustering aims to solve the non-Euclidean clustering problems, it can not well serve the clustering of web bots with complex distribution patterns.

In contrast, our method can handle the data with various complex distributions and significantly improve clustering. The average classification accuracy is about 90%, when directly using the hidden features from the Encoder block of the trained Autoencoder model. It yields 24.5% higher performance than c-mean and 30% higher accuracy than Meanshift. Furthermore, the polar coordinates strategy has significantly improved the average classification performance in comparison to directly using the hidden features, yielding about 9.5% higher accuracy. Therefore, our proposed method, extracting similar hidden features from the raw data with complex distribution patterns is a promising direction for web bot detection.

6 Conclusion

In this research, we proposed the Autoencoder-based web bot detection method, which is used to extract the hidden features of HTTP access data. The experiment results showed that our method can successfully generate those hidden features embedding and significantly distinguished the HTTP access behaviours of humans and web bots. In comparison, our method outperformed the well-known clustering methods, which yielded about 30% in web bot detection accuracy. Therefore, extracting the hidden features of various HTTP access features is important in web bot detection. In future work, we will continue the research on automatically solving the geometric angular uncertainty problem of hidden features when embedded in a low-dimensional space.

Acknowledgement. This research was supported by Webjet Limited, the company has provided valuable raw data to the research, those data are first-hand and were collected in the year of this research. It has made a contribution to this research and would be meaningful to the community.

References

1. Geroimenko, V.: Dictionary of XML Technologies and the Semantic Web, vol. 1. Springer, Cham (2004), https://doi.org/10.1007/978-0-85729-376-3
2. Menczer, F., Pant, G., Srinivasan, P., Ruiz, M.E.: Evaluating topic-driven web crawlers. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 241–249 (2001)
3. Shemshadi, A., Sheng, Q.Z., Qin, Y.: ThingSeek: a crawler and search engine for the internet of things. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1149–1152 (2016)
4. Li, X., Azad, B.A., Rahmati, A., Nikiforakis, N.: Good bot, bad bot: characterizing automated browsing activity. In: 2021 IEEE Symposium on Security and Privacy (sp), pp. 1589–1605. IEEE (2021)

5. Nagaraja, S., Shah, R.: Clicktok: click fraud detection using traffic analysis. In: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks, pp. 105–116 (2019)
6. Wang, X., Gu, B., Qu, Y., Ren, Y., Xiang, Y., Gao, L.: Reliable customized privacy-preserving in fog computing. In: ICC 2020–2020 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2020)
7. Imperva. 2022 imperva bad bot report (2018). https://www.imperva.com/resources/reports/2022-Imperva-Bad-Bot-Report.pdf
8. Basso, A., Bergadano, F.: Anti-bot strategies based on human interactive proofs. In: Stavroulakis, P., Stamp, M. (eds.) Handbook of Information and Communication Security, pp. 273–291. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-04117-4_15
9. Basso, A.: Protecting web resources from massive automated access. University of Torino, Technical RT114/08 (2008)
10. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_18
11. Jonker, H., Krumnow, B., Vlot, G.: Fingerprint surface-based detection of web bot detectors. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) ESORICS 2019. LNCS, vol. 11736, pp. 586–605. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29962-0_28
12. Rovetta, S., Suchacka, G., Masulli, F.: Bot recognition in a web store: an approach based on unsupervised learning. J. Netw. Comput. Appl. **157**, 102577 (2020)
13. Suchacka, G., Cabri, A., Rovetta, S., Masulli, F.: Efficient on-the-fly web bot detection. Knowl. Based Syst. **223**, 107074 (2021)
14. Rocha, E.: 2018 bad bot report: the year bad bots went mainstream (2018). https://www.globaldots.com/resources/blog/2018-bad-bot-report-the-year-bad-bots-went-mainstream/
15. Wang, W., Huang, Y., Wang, Y., Wang, L.: Generalized autoencoder: a neural network framework for dimensionality reduction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 490–497 (2014)
16. Yuan, X., Li, C., Li, X.: DeepDefense: identifying DDoS attack via deep learning. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–8. IEEE (2017)
17. Cabri, A., Suchacka, G., Rovetta, S., Masulli, F.: Online web bot detection using a sequential classification approach. In: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1536–1540. IEEE (2018)
18. Luo, Y., She, G., Cheng, P., Xiong, Y.: BotGraph: web bot detection based on sitemap. arXiv preprint arXiv:1903.08074 (2019)
19. Acarali, D., Rajarajan, M., Komninos, N., Herwono, I.: Survey of approaches and features for the identification of http-based botnet traffic. J. Netw. Comput. Appl. **76**, 1–15 (2016)
20. Chavoshi, N., Hamooni, H., Mueen, A.: Temporal patterns in bot activities. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 1601–1606 (2017)
21. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)

22. Danielsson, P.-E.: Euclidean distance mapping. Comput. Graph. Image Process. **14**(3), 227–248 (1980)
23. Murtagh, F., Legendre, P.: Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? J. Classif. **31**(3), 274–295 (2014)
24. Doran, D., Gokhale, S.S.: An integrated method for real time and offline web robot detection. Expert Syst. **33**(6), 592–606 (2016)
25. Rovetta, S., Cabri, A., Masulli, F., Suchacka, G.: Bot or not? a case study on bot recognition from web session logs. In: Esposito, A., Faundez-Zanuy, M., Morabito, F.C., Pasero, E. (eds.) WIRN 2017 2017. SIST, vol. 103, pp. 197–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-95095-2_19
26. Zabihimayvan, M., Sadeghi, R., Rude, H.N., Doran, D.: A soft computing approach for benign and malicious web robot detection. Expert Syst. Appl. **87**, 129–140 (2017)
27. Berners-Lee, T., Fielding, R., Frystyk, H.: Hypertext transfer protocol-http/1.0. Technical report (1996)
28. KR Suneetha and Raghuraman Krishnamoorthi: Identifying user behavior by analyzing web server access log file. IJCSNS Int. J. Comput. Sci. Netw. Secur. **9**(4), 327–332 (2009)
29. Yadav, J., Sharma, M.: A review of k-mean algorithm. Int. J. Eng. Trends Technol. **4**(7), 2972–2976 (2013)
30. Chowdhary, C.L., Acharjya, D. P.: Clustering algorithm in possibilistic exponential fuzzy C-mean segmenting medical images. In: Journal of Biomimetics, Biomaterials and Biomedical Engineering, vol. 30, pp. 12–23. Trans Tech Publications Ltd (2017)
31. Derpanis, K.G.: Mean shift clustering. Lect. Notes **32**, 1–4 (2005)