Applied Soft Computing 13 (2013) 698–708

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Detection of malicious and non-malicious website visitors using unsupervised neural network learning

Dusan Stevanovic*, Natalija Vlajic, Aijun An

Department of Computer Science and Engineering, York University, 4700 Keele St., Toronto, Ontario, M3J 1P3, Canada

ARTICLE INFO

Article history:
Received 8 August 2011
Received in revised form 24 April 2012
Accepted 6 August 2012
Available online 23 August 2012

Keywords:
Web crawler detection
Neural networks
Web server access logs
Machine learning
Clustering
Denial of service

ABSTRACT

Distributed denials of service (DDoS) attacks are recognized as one of the most damaging attacks on the Internet security today. Recently, malicious web crawlers have been used to execute automated DDoS attacks on web sites across the WWW. In this study, we examine the use of two unsupervised neural network (NN) learning algorithms for the purpose web-log analysis: the Self-Organizing Map (SOM) and Modified Adaptive Resonance Theory 2 (Modified ART2). In particular, through the use of SOM and modified ART2, our work aims to obtain a better insight into the types and distribution of visitors to a public web-site based on their browsing behavior, as well as to investigate the relative differences and/or similarities between malicious web crawlers and other non-malicious visitor groups. The results of our study show that, even though there is a pretty clear separation between malicious web-crawlers and other visitor groups, 52% of malicious crawlers exhibit very 'human-like' browsing behavior and as such pose a particular challenge for future web-site security systems. Also, we show that some of the feature values of malicious crawlers that exhibit very 'human-like' browsing behavior are not significantly different than the features values of human visitors. Additionally, we show that Google, MSN and Yahoo crawlers exhibit distinct crawling behavior.

1. Introduction

The today's business world is critically dependent on the availability of Internet. For instance, the phenomenal growth and success of Internet has transformed the way traditional essential services, such as banking, transportation, medicine, education and defence, are operated. In ever increasing numbers, these services are being offered by means of web-based applications. However, the inherent vulnerabilities of the Internet architecture provide opportunities for various attacks on the security of web-based applications. Distributed denial-of-service (DDoS) is an especially potent type of security attack, capable of severely degrading the response-rate and quality at which web-based services are offered. According to the United States' Department of Defence report from 2008, presented in Ref. [1], the number of cyber attacks (including the DDoS attacks) from individuals and countries, targeting economic, political and military organizations, are expected to increase in the future and cost billions of dollars.

The most common way of conducting a denial of service (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. In the past, most DoS attacks were single-sourced, which means they were reasonably easy to prevent by locating and disabling the source of the malicious traffic. Nowadays, however, almost all DoS attacks involve a complex, distributed network of attacking machines – comprising thousands to millions of hijacked zombies. These, the so-called DDoS attacks, are extremely difficult to detect due to the sheer number of hosts participating in the attack. At the same time, they can generate enormous amount of traffic toward the victim and result in substantial loss of service and revenue for businesses under the attack.

An emerging (and increasingly more prevalent) set of DDoS attacks, known as *application-layer* or *layer-7* attacks [2], are shown to be particularly challenging to detect. The reasons for this are: (1) in an application-layer attack, the attacker utilizes a legitimate-looking layer-7 network session, and (2) HTML requests sent to a web server are often conducted by a cleverly programmed crawler,[1]

* Corresponding author. Tel.: +1 416 736 2100x70143.
E-mail address: dusan@cse.yorku.ca (D. Stevanovic).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.08.028

[1] *Web-crawlers* are programs that traverse the Internet autonomously, starting from a *seed* list of web pages and then recursively visiting documents accessible from that list. Crawlers are also referred to as *robots* (*bots*), *wanderers*, *spiders*, or *harvesters*. Their primary purpose is to discover and retrieve content and knowledge from the web on behalf of various web-based systems and services. For instance, search-engine crawlers seek to harvest as much web content as possible on a regular basis, in order to build and maintain large search indexes. On the other hand, shopping bots

in a way that mimics a semi-random walk through the web site links, and thus appears as a web site traversal conducted by a legitimate human user. Given the fact that application-layer DDoS attacks resemble the legitimate traffic, it is quite challenging not only to defend against these attacks but also to construct an effective metric for their detection.

So far, a number of studies on the topic of application-layer DDoS attacks have been reported. Thematically, these studies can be grouped into two main categories: (1) detection of application-layer DDoS attacks during a *flash crowd* event based on aggregate-traffic analysis [3,4] and (2) differentiation between well-behaved and malicious web crawlers based on web-log analysis [5–7].

The study presented in this paper falls in the latter of the above mentioned categories, as we examine the use of two unsupervised neural network (NN) learning algorithms for the purpose web-log analysis: the Self-Organizing Map (SOM) [8] and Modified Adaptive Resonance Theory 2 (Modified ART2)[2] [9]. In particular, through the use of SOM and ART2, our work aims to obtain a better insight into the types and distribution of visitors to a public web-site based on their browsing behavior, as well as to investigate the relative differences and/or similarities between malicious web crawlers and other non-malicious visitor groups.

In our earlier work [10], we have investigated the use of supervised algorithms (as provided by WEKA data-mining software [11]) for the purpose of web-user classification, including: C4.5, RIPPER, k-Nearest Neighbours, Naïve Bayesian Learning, Support Vector Machine. The results of this study have shown that supervised classification of web-users into different visitor categories (malicious vs. well-behaved vs. unknown visitors) can be effective and ensure satisfactory levels of classification accuracy, but only if preceded by a reliable data-labelling process. Namely, the main known disadvantage of most supervised algorithm, including those studied in [10], is the fact that they are only as good as their respective data-labelling strategy. Put another way, a supervised algorithm can provide accurate classification only if it has been trained on correctly labelled data, which in turn requires that the human/labelling expert be very familiar with the type and nature of the data-set being studied. Unfortunately, in the emerging era of highly sophisticated and ever-evolving web crawlers and bots (i.e., crawlers and bots that aim to hide or fake their identity by mimicking the behaviour of regular human visitors), the use of 'expert knowledge' for the purpose of reliable data pre-classification/labelling will be increasingly more problematic. Clearly, from the perspective or web-user classification, the presence of crawlers and bots with dynamically changing human-like behaviour is likely to translate into highly irregular and noisy data, and as such present a great challenge for any supervised expert-based system. This, ultimately, explains our motivation to extend the work presented in [10] and look at use of unsupervised learning for the purpose of web-user classification.

The content of this paper is organized as follows: in Section 2, we discuss previous works on web crawler detection. In Section 3, we give an overview of our web-log analyzer that is used to generate a meaningful training dataset out of any given access log file. In Section 4, we briefly outline our experimentation setup. In Section 5, we present and discuss the obtained web-session clustering results. In Section 6, we conclude the paper with final remarks.

2. Related work

To date, in addition to our work [10], several other research studies have also looked at the use of supervised learning for the purposes of data-mining and/or clustering of web sessions. Note that supervised learning process clusters sessions based on previous a priori knowledge. In one of the first such studies [12], the authors attempt to discover distinct groups of web robot sessions by applying C.4.5 algorithm (i.e., a decision tree classifier) to 25-dimensional feature vector space. The 25 features, i.e., their respective values, are derived from the navigational properties of each identified robot session. In advance of clustering, and depending on the value of *user-agent fields*, each session is pre-labeled as known *robots*, *known browsers*, *possible robots*, and *possible browsers*. The results of the study show that, by applying the proposed feature set in combination with C.4.5 algorithm, robots can be detected with more than 90% accuracy after only four web-page requests. In [13], the authors utilize supervised Bayesian classifier to detect the presence of web crawlers from web server logs and, subsequently, they compare their results to the results obtained with the decision tree technique. The proposed methodology achieves very high recall and precision values in web robot detection. Another study utilizing logistic regression and decision trees has been reported in [6]. In this study, authors propose a robot detection tool that speeds up the tasks for pre-processing web server access logs and achieves very accurate web robot detection.

Several studies have looked at the use of unsupervised learning for the purpose of more general web log analysis. In [14], the authors employ the Self-Organizing Map (SOM) algorithm to achieve automatic demographic-based classification of human web-site visitors based on the number and sequence of their web-page visits. In [15], the authors also examine the application of the SOM algorithm on web-server access logs, with the aim to group human web-visitors thematically and, as a result of that, help them find relevant information in a shorter period of time. In a similar study [16], the authors propose employing the Adaptive Resonance Theory (ART) algorithms to cluster human web users according to their thematic interests.

In the view of the previous works, the novelty of our research is twofold:

1) Firstly, to the best of our knowledge, this is the first study that applies unsupervised learning to the problem of web-visitor categorization, ultimately aiming to promote effective differentiation between malicious web-crawlers and other (non-malicious) visitor groups to a web site. (Note, in [14–16], only human web-visitors have been considered, and little to no attention has been given to automated web-crawlers.) We have chosen to use the SOM and ART neural network algorithms in our study for the following reasons.
   - The goal of the SOM algorithm is to cause the underlying neural network to respond similarly to similar input patterns. In order to achieve this goal, the network undergoes multiple rounds of the so-called *competitive learning process*. (In competitive learning, a training sample is fed to the network, and its Euclidean distance to all weight vectors is computed. The neuron with weight vector closest to the training sample is pronounced 'winner'. Once identified, the weight vector of the winner, as well as a number of its nearest topological neighbors, are adjusted towards the given training sample. The

crawl the web to compare prices and products sold by different e-commerce sites. Malicious crawlers are type of web robots that, for instance, generate DDoS traffic that can overwhelm web server's resources and thus limit or unable legitimate users' access to the website. Another example of malicious activity attributed to malicious crawlers is collecting email addresses for spam mail.

[2] Modified ART2 is a variation of the original ART algorithm [24]. Its advantages over the original algorithm are: (1) stable learning that results in gradually increasing/merging clusters, and (2) learning/clustering that can be terminated either when the radius of the formed clusters reaches some predetermined size, or when the number of formed clusters reaches some predetermined number.

magnitude of adjustment decreases with the number of learning epochs and with the distance from the winning node. For more on the SOM algorithm see [17]). From the practical point of view, the SOM algorithm is well known for: a) its *topology preservation* ability, which implies that similar input samples activate topologically close neurons, b) its ability to produce natural clustering, i.e. clustering that is robust to statistical anomalies, and c) superior visualisation of high-dimensional input data in 2D-representation space.

- The ART2 algorithm also deploys the concept of competitive learning, but in combination with the so-called *winner-takes-all* rule, ultimately producing fundamentally different clustering results. Namely, as in the case of SOM, a training sample is fed to the ART2 network, and its respective winning neuron is found by means of minimum Euclidean distance. However, unlike the SOM, ART2 proceeds with the adjustment of the winner's weight vector only if it is deemed sufficiently close to the training sample (i.e., falls within the so-called *vigilance threshold*). If the vigilance threshold criterion is not met, other best-matching neurons are checked. If no neuron committed to the network so far is found to satisfy the threshold criterion, then a new uncommitted neuron is added and adjusted towards the training sample. For more on the ART2 algorithm see [18]. From the practical point of view, the ART2 algorithm is known for: a) its ability to preserve the balance between retaining previously learned patterns and learning new ones, the so-called *stability-plasticity property*; and b) its ability to identify statistically underrepresented but significant clusters, thus being greatly suited for imbalanced datasets.

2) Second novelty is the fact that, to the best of our knowledge, this is the first study that attempts to examine the actual, qualitative differences between malicious web-crawlers and other non-malicious crawler types, such as GoogleBot and MSNBot, by applying the SOM-based data visualisation methods.

3. Pre-processing of server logs

In our study, a Java-based log analyzer has been utilized to pre-process the web server access-log files. A typical web server access log file includes the information such as the IP address/host name of the site visitor, the URL of requested page, the date and time of the request, the size of the data requested and the HTTP method of request. Additionally, the log contains the user agent string describing the hardware and/or software the visitor was using to access the site, and the referrer field which specifies the web page by which the client reached the current page.

On each provided access log file, our log analyzer performs the following: (1) scans the entries in the log to identify unique visitor sessions, and (2) for each identified session, the analyzer examines its key features to generate the sessions' 10-dimensional feature-vector representation.

In the reminder of this section, we provide a detailed description of the above mentioned processes (session identification and generation of sessions' feature-vector representations) as well as the process of dataset labeling, as performed by our log analyser and illustrated in Fig. 1. Note from Fig. 1 that the aggregate of the obtained feature-vectors comprises the training and testing datasets that are to be used for training and evaluation of the SOM and Modified ART2 algorithm. We close this section with the description of dataset labeling process, which forms a critical step in enabling meaningful validation of the results of the clustering process.

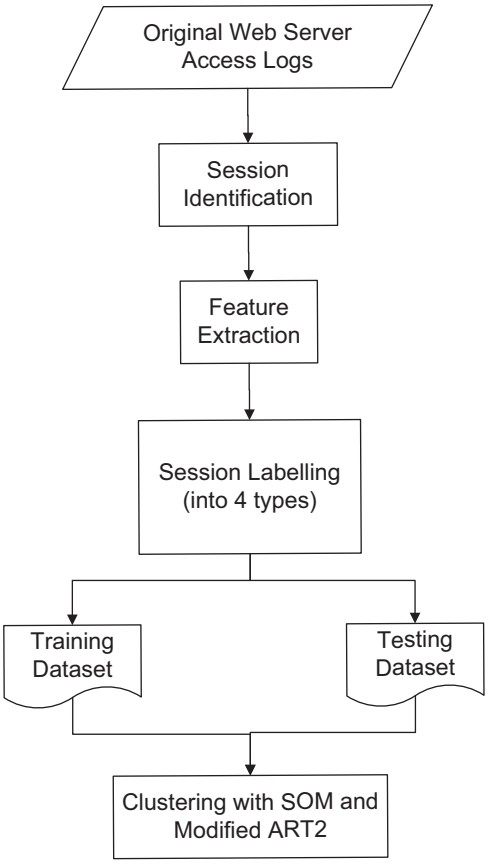


Fig. 1. Web server access log pre-processing.

3.1. Session identification

Session identification is the process of dividing a server access log into sessions. Typically, session identification is performed by: (1) grouping of all HTTP requests that originate from the same IP address and are described by the same user-agent string, and (2) by applying a timeout approach to break this grouping into unique sessions. Therefore, a session is defined as a sequence of requests coming from the same IP address (and is described by the same user-agent string) and where the time-lapse between any two consecutive HTTP requests in the sequence is within a pre-defined threshold. The key challenge of session identification is to determine the proper value of the given threshold, as different web users exhibit different navigational behavior. In the majority of web-related literature, 30-min period has been used as the most appropriate maximum session length (see Ref. [13]). Hence, our log analyser employs the same 30-min threshold to distinguish between different sessions launched by the same user.

3.2. Features

From previous studies on web session analysis, namely [6,12,13,19] we have adopted eight different features that are shown to be useful in identifying and distinguishing between automated and human visitors to a web site. These features are enlisted below:

1. Click number – a *numerical* attribute calculated as the number of HTTP requests sent by a user in a single session. The click number metric appears to be useful in detecting the presence of the web crawlers because higher click number can only be

achieved by an automated script (such as a web robot) and is usually very low for a human visitor.

2. HTML-to-image ratio – a *numerical* attribute calculated as the number of HTML page requests over the number of image file (JPEG and PNG) requests sent in a 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.
3. Percentage of PDF/PS file requests – a *numerical* attribute calculated as the percentage of PDF/PS file requests sent in a single session. In contrast to image requests, some crawlers, tend to have a higher percentage of PDF/PS requests than human visitors, e.g., a crawler traversing through a site would typically attempt to retrieve all encountered PDF/PS files, while a human visitor would be much more selective about what he chooses to retrieve.
4. Percentage of 4xx error responses – 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.
5. Percentage of HTTP requests of type HEAD – a *numerical* attribute calculated as percentage of requests of HTTP type HEAD sent in a single session. (In the case of an HTTP HEAD request, the server returns the response header only, and not the actual source, i.e., file.) 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.
6. Percentage of requests with unassigned referrers – a *numerical* attribute calculated as the percentage of blank or unassigned referrer fields set by a user in a single session. Most web crawlers initiate HTTP requests with unassigned referrer field, while most browsers provide referrer information by default.
7. Number of bytes requested from the server – a *numerical* attribute calculated as the amount of data, in bytes, that was requested from the server in a single session. Typically, sessions belonging to web robots should request greater amounts of data from the server in a single session than sessions initiated by human visitors.
8. Page Popularity Index – a *numerical* attribute, calculated as the average value of Page Popularity Index (PPIs) for all N pages (Page(i), $i = 1, \ldots, N$) retrieved during one observed session j – PPI_Session(j). The expression for PPI_Session(j) is given in Eq. (1):

$$\text{PPI\_Session}(j)\frac{\sum_{i=1}^{N}\{(\max(\text{PPI})-\text{PPI}(i))\text{number of Session}(j)\text{ requests for Page}(i)\}}{\text{Total number of Session}(j)\text{ page requests}} \quad (1)$$

The expression for the Page Popularity Index for a page i – PPI(i) in the above expression – is shown in Eq. (2).

$$\text{PPI}(i) = -\log\left(\frac{\text{Number of requests for Page}(i)}{\text{Total number of requests}}\right) \quad (2)$$

Note, in Eq. (1), only requests within/for one particular session are considered. In Eq. (2), the overall/total number of requests is considered – cumulatively, from all sessions. It should be obvious from Eqs. (1) and (2) that in order to calculate PPI_Session(j), first PPI(i) for all pages appearing in the log file needs to be calculated. Also, note that the max(PPI) is the maximum PPI out of all pages. The more detail discussion of the page popularity index and its calculation can be found in Ref. [19]. In general, it is reasonable to assume that human visitors would request more popular pages in a single session and therefore would have a higher page popularity index score while web robots would request both popular and unpopular pages which would result in a lower page popularity index score for their respective sessions.

As mentioned earlier, features 1–8 have been used in the past for distinguishing between human- and robot-initiated sessions. However, based on the recommendations and discussion presented in Ref. [20], we introduce two novel features for characterization of web-browsing sessions:

9. Standard deviation of requested page's depth – a *numerical* attribute calculated as the standard deviation of page depth across all requests sent in a single session. For instance, we assign a depth of three to a web page '/cshome/courses/index.html' and a depth of two to a web page '/cshome/calendar.html'.
10. Percentage of consecutive sequential HTTP requests – a *numerical* attribute calculated as the percentage of sequential requests for pages belonging to the same web directory and generated during a single user session. For instance, a series of requests for web pages matching pattern '/cshome/course/*.*' will be marked as consecutive sequential HTTP requests. However, a request to web page '/cshome/index.html' followed by a request to a web page 'cshome/courses/index.html' will not be marked as consecutive sequential requests.

The importance of features 9 and 10 can be explained as follows:

In a typical web-browsing session, humans are set to find information of interest by following a series of thematically correlated and progressively more specific links. In most web-server systems, thematically correlated information (i.e., respective files) tends to be stored at the same or similar depth in the file system hierarchy. In contrast, robots' browsing sessions tend to be far more extensive. Namely, most web crawlers (e.g., Google, Yahoo, etc.), browse systematically through an entire web-domain and access files at various depths in the file hierarchy. Based on the above, it is reasonable to assume that the standard deviation of requested pages' depths, i.e., attribute 8, remains low for most sessions belonging to human visitors, and high for most web robot sessions.

Also the number of resources requested in a single session is another distinction between robot and human traffic that is not expected to change over time. This distinction arises from the fact that human users retrieve information from the Web via some interface, such as a web browser. This interface forces the user's session to request additional resources automatically. Most Web browsers, for example, retrieve the HTML page, parse through it, and then send a barrage of requests to the server for embedded resources on the page such as images, videos, and client side scripts to execute. Thus, the temporal resource request patterns of human visitors are best represented as short bursts of a large volume of requests followed by a period of little activity. In contrast, web robots are able to make their own decisions about what resources linked on an HTML page to request, and may choose to execute the scripts available on a site only if they have the capacity to do so. Based on the above arguments, it is reasonable to expect that the number of consecutive sequential HTTP requests would be relatively high in human user sessions and relatively low in web robot sessions.

3.3. Dataset labeling

Once the training dataset (comprising feature-vector representations) is generated, the log analyzer labels each feature-vector

as belonging to one of the following 4 categories: *human visitors*, *well-behaved web crawlers*, *malicious crawlers* and *unknown visitors*. The goal of data labeling is to facilitate our understanding and validation of the results that are to be obtained by the actual clustering process. Namely, through quick association of feature-vectors corresponding to a cluster with their pre-assigned labels, we hope to be able to obtain a better understanding of the cluster's nature and significance.

In our work, the labeling of feature vectors is performed by observing whether the respective user has attempted to access the *robots.txt* file. Namely, web administrators use a special-format file called *robots.txt* to instruct the visiting robots about parts of their sites that should NOT be visited by an automated crawler. (For example, when a robot visits a web-site, say http://www.cse.yorku.ca, it should first check for http://www.cse.yorku.ca/robots.txt to learn about the site rules.) It is unlikely that any human would check for robots.txt, since there are no external or internal hyperlinks leading to this file, nor are (most) users aware of its existence. Therefore, in our study, any web session that attempts to access the robots.txt file is considered to be generated by an automated program (i.e., a crawler). The actual algorithmic steps of the labeling process are outlined below:

(1) Any feature vector that corresponds to a web session whose user agent string matches a known browser and does not access the 'robots.txt' file is labeled as *human visitors*.
(2) Any feature vector that corresponds to a web session whose user agent string matches a known well-behaved web crawler is labeled as *well-behaved web crawlers*.
(3) Any feature vector that corresponds to a web session whose user agent string matches a known malicious web crawler is placed in a cluster of *malicious web crawlers*. Also any session belonging to a human visitor or unknown visitor in which the user accesses the 'robots.txt' file is also placed in a cluster of *malicious web crawlers*.
(4) All other web sessions are labeled as *unknown visitors*.

Note, the log analyzer maintains a table of user agent fields of all known (malicious or well-behaved) web crawlers. This table can be built from the data found on web sites [21,22]. The web sites also maintain the list of various browsers' user agent strings that can be used to identify human visitors to the site as well.

4. Experimental design

4.1. Training data

In the experimental stage of our study, the training data sets were constructed by pre-processing web server access log files provided by York University's computer science and engineering department. The log files contained detailed information about user web-based access into the domain www.cse.yorku.ca during a 4-week interval – between mid April 2011 and mid May 2011. A total of about 3 million log entries were extracted from the file. Table 1 lists the number of sessions and class label distributions generated by the log analyzer.

Table 1
Class distribution in the dataset.

	Number of sessions
Total #	65,576
Total # of human sessions	53,640
Total # of well-behaved crawler sessions	7607
Total # of malicious crawler sessions	287
Total # of unknown visitor sessions	4042

4.2. Clustering algorithms

The detection of web crawlers was evaluated with the following two unsupervised neural network algorithms: SOM and ART2. The evaluation using the SOM relied on the algorithm's implementation contained in MATLAB's Neural Network Toolbox. We chose a SOM comprising 100 neurons in 10-by-10 hexagonal arrangement. The map was trained with 200 epochs. The Modified ART2 was implemented in MATLAB following the pseudo-code outlined in Ref. [9]. The algorithm parameters were set to: $\rho_{max} = 1.5$, $\Delta\rho = 0.1$ and $n_{max} = 5$. All input vectors were normalized prior to being fed to SOM and Modified ART2.

5. Clustering results

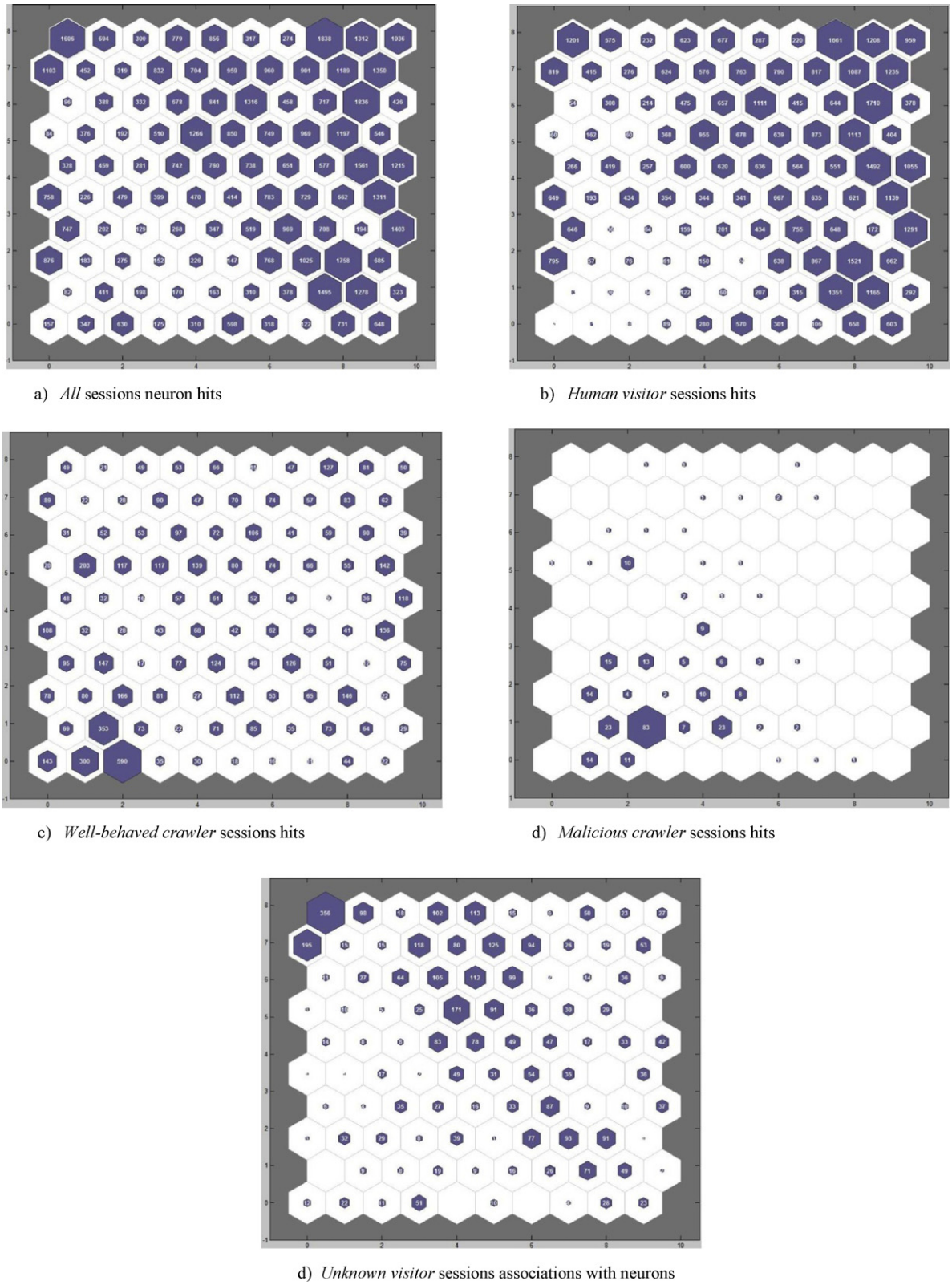
5.1. SOM results

Fig. 2 displays the results of dataset clustering obtained with a 10-by-10 neuron SOM. On each of the shown maps, the size of the blue region inside a neuron's hexagon depicts the number of session hits for that neuron, i.e., number of sessions whose feature vectors end up firing the (same) given neuron. The exact number of a neuron's session hits is also explicitly provided within the neuron's hexagon region.

The map in Fig. 2a shows the neuron hits for all sessions, and thus helps us visualize the actual distribution of the training dataset (i.e., helps us get an idea about the number, size and spatial proximity of the dataset's most dominant clusters). Fig. 2b–e shows the neuron hits for sessions that were pre-labeled as belonging to human, well-behaved crawler, malicious crawler and unknown visitors, respectively. From the obtained maps, the following interesting conclusions can be drawn:

- *Human* vs. *crawlers sessions*: Based on the distribution of fired neurons in Fig. 2b–d, there appears to be a reasonably good separation between human visitor sessions and web-crawler sessions (both malicious and well-behaved). Namely, while crawler sessions are almost exclusively associated with neurons in the lower left corner of the map, human sessions are spread over a large area of the map, with most human sessions firing the neurons in the upper right corner of the map. It might be worth pointing out that the large spread of fired neurons in the map of Fig. 2b is *not* an indicator of greater variability in humans sessions compared to other session groups. Instead, it is the result of the statistical dominance of training-data corresponding to human sessions – see Table 1. (As indicated in the introduction, the SOM algorithm produces results that are dependent on the input data density; hence, data clusters with higher density tend to 'win-over' a larger number of SOM neurons, regardless of their inter-cluster variance.)
- *Sessions that are labeled as human but 'behave' like malicious crawlers:* A detailed inspection of Fig. 2b and d reveals that, in spite of the well-formed separation between human and malicious web-crawlers, a percentage of sessions/visitors that declare themselves as regular (human) visitors[3] end up firing neurons in the region (or close to the region) dominated by malicious web-crawlers – lower left corner of the map. In the view of our data-labeling algorithm from Section 3.3, this observation raises the question whether those sessions, in fact, correspond to malicious crawlers whose aim is to bypass web-site security by simply not accessing the robots.txt file and/or falsifying the value of user

[3] Recall from Section 3.3 that we consider all sessions that carry the name of a well-known browser in the user agent field and that do not access the robots.txt to be human sessions.



a) *All* sessions neuron hits

b) *Human visitor* sessions hits

c) *Well-behaved crawler* sessions hits

d) *Malicious crawler* sessions hits

d) *Unknown visitor* sessions associations with neurons

Fig. 2. Session hits per neuron visualized in 2-dimensional SOM map. (a) *All* sessions neuron hits; (b) *Human visitor* sessions hits; (c) *Well-behaved crawler* sessions hits; (d) *Malicious crawler* sessions hits; (e) *Unknown visitor* sessions associations with neurons.

agent string field. Recall, the initial accessing of the robots.txt file would ideally be performed by automated crawlers visiting a web site, but cannot be enforced. Similarly, user agent string appears as a parameter in HTML requests, and can be relatively easily altered.

- *Sessions that are labeled as malicious crawlers but 'behave' like humans:* A detailed inspection of Fig. 2b and d also reveals that a number of sessions/visitors that are identified as malicious crawlers end up firing neurons in the region dominated by human generated sessions – upper part of the map. It is reasonable to

assume that these visitors are indeed malicious crawlers, as it is unlikely that any regular human visitor would change the value of its agent string into 'malicious crawler', thus risking to be blocked by the web-site. Accordingly, this observation implies that the behavior of some malicious crawlers – those that fire the nodes in the upper part of the map – is very similar to the behavior of regular users. It should be obvious that such malicious crawlers are potentially very dangerous. Namely, had they attempted to falsify the value of agent string (i.e., declare themselves as regular visitors), they would have 'perfectly' blended into the population of regular human visitors, and would be very hard to detect by the web-site's security system.

- *Unknown visitor sessions*: As explained in Section 3.3, unknown visitor sessions are sessions whose user agent strings are not known and do not access the robots.txt file and thus are not enlisted on [21,22]. By comparing Fig. 2b and e, it is interesting to observe that there is a significant overlap between fired neurons in the respective maps. This leads us to conclude that most unknown sessions are likely generated by regular human users, i.e., are likely non-malicious by their behavior and intent.

5.2. Modified ART2 results

Fig. 3 displays the results of dataset clustering using Modified ART2. The plot displays the ratio of each session type (human, well-behaved crawler, malicious crawler and unknown visitor) per cluster placement. A session is placed in cluster *i*, if its 10-dimensional vector representation is the closest (measured in the Euclidean distance) to the centroid of cluster *i* among all other clusters. The plot displays the sample results when Modified ART2 algorithm generates 5 clusters of sessions.

While the results generated with SOM are useful for obtaining information about the spatial distribution, i.e., proximity, of data clusters, Modified ART2 gives us an insight into the inter-cluster variance. (As indicated in the introduction, Modified ART2 creates equal-size clusters and is not influenced by statistical irregularities in the training dataset.) With this in mind, and by expecting Fig. 3, we derive the following conclusions:

- *Human sessions*: Nearly 92% of human sessions fall into cluster 1 or cluster 3, thus suggesting a relatively small behavioral variance of this user group. In practical terms, this implies that human users tend to follow very similar web browsing patterns.
- *Unknown sessions*: Over 80% of unknown sessions belong to the same clusters as human sessions, namely clusters 1 and 3. This confirms our hypothesis from Section 5.1, that most unknown sessions are likely human-generated.
- *Malicious web-crawler sessions*: Out of all session groups, malicious crawlers exhibit the greatest variability – they are spread over all 5 formed clusters, with most being assigned to cluster 3. It is interesting to observe that nearly 14 and 38% of malicious web-crawler sessions are assigned to clusters 1 and 3, respectively, together with human visitors. This again confirms our earlier hypothesis, that some malicious web crawlers behave very-much like regular users, and in the case of a falsified user agent string value their detection would have presented a particular challenge.

5.3. Outlier analysis

In the preceding section, we have identified two groups of sessions that could present a particular challenge for web intrusion detection systems: (1) sessions that are labeled as malicious crawlers but 'behave' like humans, and (2) sessions that are labeled as humans but 'behave' like malicious crawlers. In this section we undertake a detailed analysis of the underlying characteristics of these two groups of sessions, which we will refer to as *outlier sessions* in the remainder of the document.

5.3.1. Outlier sessions that are labeled as malicious crawlers but 'behave' like humans

In order to gain a better understanding of the malicious crawler sessions that 'behave' like humans, we have first identified and grouped together all malicious crawler sessions belonging to the upper half of the map in Fig. 2d and are associated with clusters 1 and 3 in Fig. 3. There has been a total of 155 such sessions. Subsequently, we have: (1) identified the user agent strings and respective source-IP addresses associated with this group of malicious sessions, and (2) applied the significance of the difference test on individual feature values of session belonging to this group and the actual human sessions – namely human sessions that belong to clusters 1 and 3 in Fig. 3 and are clustered in the upper half of the map in Fig. 2b.

5.3.1.1. (a) User agent strings and source-IP addresses of malicious crawlers that 'behave' like humans. Table 2 displays the user agent strings of malicious outlier sessions together with their respective source IP addresses, as well as the overall number of sessions that have employed the given user agent strings. According to information posted on http://www.botsvsbrowsers.com/ and http://www.useragentstring.com websites, these sessions are either known as malicious (e.g., sogou web crawler) or belong to unknown bots that access the robots.txt file (all but the sogou web crawler).

Also note in Table 2 that 38 (24.5%) of the malicious crawler outlier sessions have a blank user agent string. At the same time, 92% of unknown sessions (presented in Fig. 2d are observed to have blank user agent string as well. As explained in Section 3.3, these sessions are labeled as not malicious and instead are placed in the unknown session group because they do not access the robots.txt file. We have observed, however, that in the SOM map some of these unknown sessions with blank user agent string turn up in the same area as malicious outlier sessions. This is a strong indication

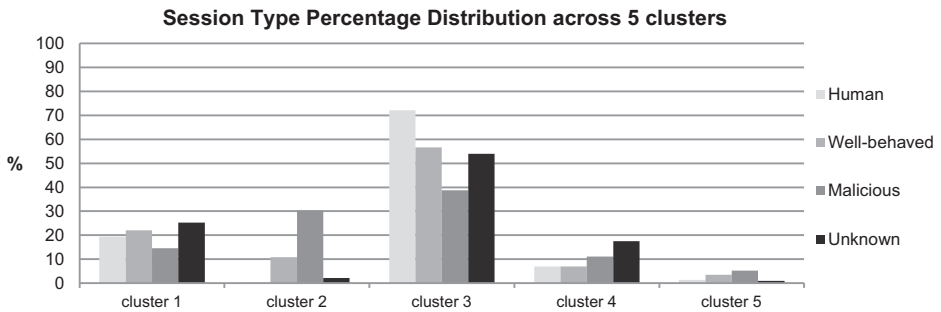


Fig. 3. Session type percentage distribution across 5 clusters.

Table 2
Common types of user agent strings among malicious crawlers that 'behave' like humans.

User agent string	Originating IP addresses	Number of malicious outlier sessions with this user agent string	Geographical location of the IP addresses
sogou	123.126.50.76, 123.126.50.69, 123.125.116.113, 220.181.94.231, 220.181.125.68, 220.181.93.3, 219.143.205.12	54	China
Empty user agent string	123.126.50.76	38	China
Valid Firefox user agent string	14.52.133.143, 128.125.163.222, 168.156.70.30	16	Korea, California, Washington, DC
Michigan papers crawler	141.211.202.150	15	Michigan
Firefox 3.0	212.227.101.236	11	United Kingdom
topyx-crawler	194.199.60.68	5	France
winhttp.winhttprequest.5	122.228.159.218, 76.94.218.142	3	China, California
linkchecker/6.9	24.89.182.19	3	New Jersey
Valid internet explorer string	24.12.142.226, 62.24.181.134, 35.11.51.17	3	Illinois, Scotland, Michigan
quickobot	110.234.114.50	2	India
dibot	188.40.99.137	2	Germany
qryos	202.71.101.68	1	Malaysia
teleport pro	69.64.84.92	1	New Hampshire
raymond balm\	91.121.119.167	1	France

that such unknown sessions are also likely malicious crawlers that are pretending to be human, but end up in the unknown group just by virtue of avoiding to access the robots.txt file. Therefore, web administrators should take a note of all sessions with blank user agent strings.

5.3.1.2. (b) Significance of the difference test. In our second experiment, we have applied the significance of the difference test on individual feature values between malicious outlier sessions that 'behave' like human and the actual human sessions.

The calculation of the significance of the difference test (i.e., *t*-test) is based on the following expression:

$$t(f_i) = \frac{|\text{mean}_1(f_i) - \text{mean}_2(f_i)|}{\sqrt{\text{Var}_1(f_i)/n_1 + \text{Var}_2(f_i)/n_2}} \quad (3)$$

In the equation above, mean_1 and mean_2 are the means of the ith feature value, Var_1 and Var_2 are the variances of the ith feature value, and n_1 and n_2 are the number of elements in the two groups of sessions, respectively. The degrees of freedom value used in the t-test is $n_1 + n_2 - 1$. Note that the values of the ith feature in the two groups are considered significantly different (with 97.5% confidence) if the t value in Eq. (3) is greater than 1.96 with degrees of freedom greater than 100. The significance of the difference test is explained in greater detail in Ref. [23].

The results of the significance of the difference test are shown in Table 3. The table also displays the mean and variance for each feature value in the two groups of sessions. Based on the overall t-test results, the malicious sessions that 'behave' like humans are shown to exhibit human-like browsing characteristics in terms of the mean value of feature 5. Put another way, the obtained result suggests that the greatest similarity between human browsing and the browsing of malicious crawlers that 'behave' like human is in the relatively small percentage of HEAD requests. Hence, it is our recommendation that web administrators pay special attention to this feature and closely inspect all web crawler sessions with a very small number of HEAD requests relative to the number full GET HTTP requests.

Table 3
Mean, variance and significance of the differences test results on feature values between actual human sessions and malicious outlier sessions.

Features	Significant difference	Mean/variance actual human sessions	Mean/variance malicious outlier sessions
Click rate	Yes	0.361/0.431	0.06/0.029
Html to image ratio	Yes	0.555/2.208	0.34/0.035
% of PDF documents	Yes	0.13/0.092	0.233/0.077
% of Error requests	Yes	0.05/0.011	0.227/0.042
% of HEAD requests	No	0.0016/0.00046	0.014/0.009
% of unassigned referrers	Yes	0.063/0.022	0.53/0.189
Number of bytes requested	Yes	5.206/1.16	4.87/1.39
Popularity Index	Yes	5.828/4.3	2.47/3.36
S.D. of page depth	Yes	0.54/0.19	0.92/0.25
% of sequential requests	Yes	0.63/0.055	0.487/0.072

5.3.2. Sessions that are labeled as human but 'behave' like malicious crawlers

Another interesting group of sessions are sessions that are labeled as human but 'behave' like malicious crawlers. Please recall, these are the sessions that end up firing neurons in the lower left corner of the map in Fig. 2b, and are also associated with clusters 2, 4 and 5 in Fig. 3. There has been a total of 4873 such sessions. Through our analysis, we have: (1) identified the common user agent strings and geographical location of source-IP addresses associated with this group of human sessions, and (2) applied the significance of the difference test on individual feature values of session belonging to this group and the actual malicious sessions - namely malicious sessions that belong to clusters 2, 4 and 5 in Fig. 3 and are clustered in the lower left corner of the map in Fig. 2b.

5.3.2.1. (a) User agent strings and geographical location of source-IP addresses of human crawlers that 'behave' like malicious. Due to

Table 4
Geographical location distribution of human sessions that 'behave' like malicious crawlers.

Country origin of IP addresses	%	City origin of IP addresses	%
Canada	25	Toronto and York University Domain	17
India	16	New Delhi	5
United States	10	Sukkur, Beijing	3
UK, China, Pakistan, Philippines	4	Madras, Windhoek, Vienna, Strathfield, Singapore	2
Kuwait, Australia	3	Other	49 (<1%)
Others	27 (<2%)	Unknown	13

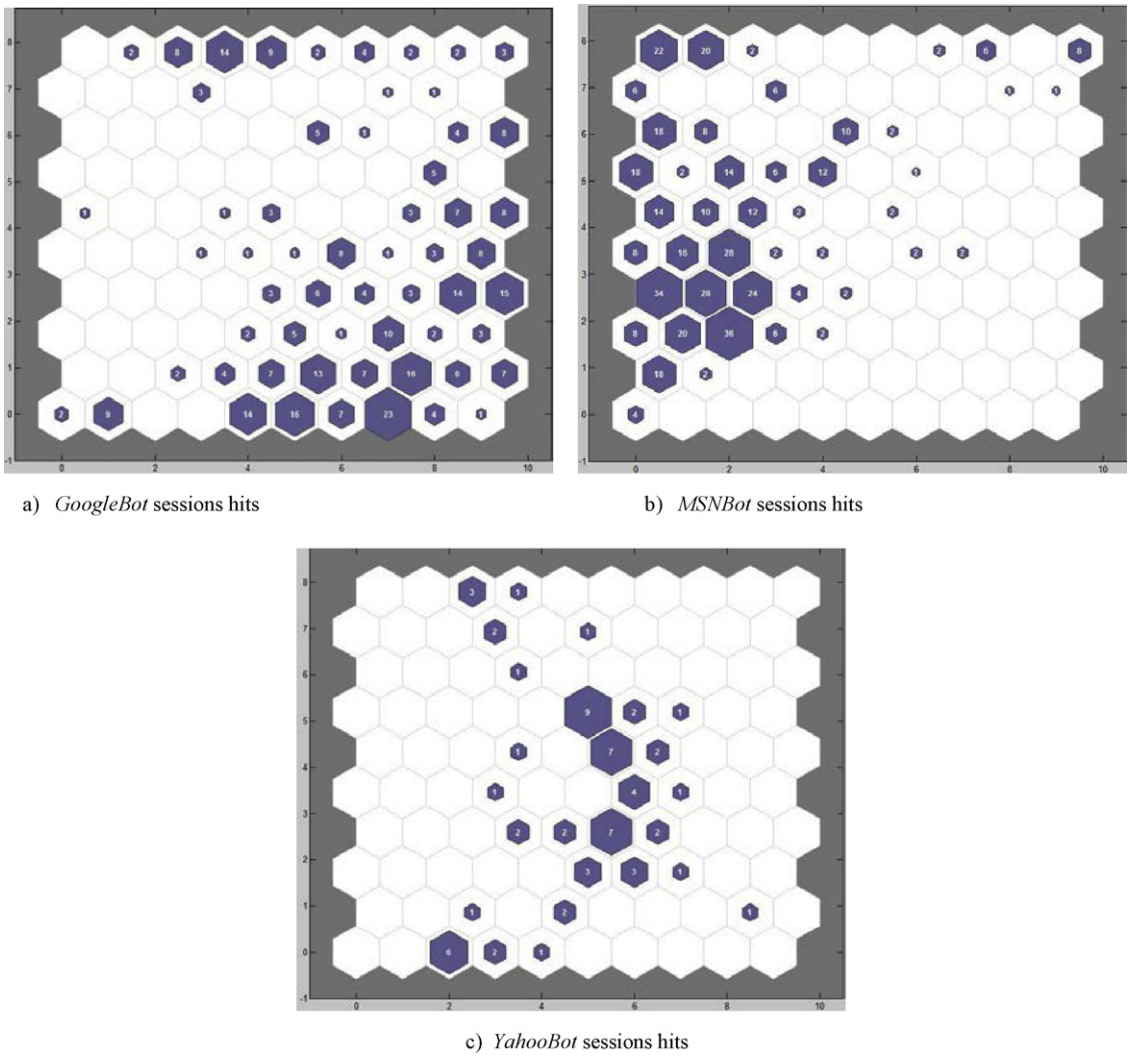
privacy reasons we cannot list the IP addresses of the most common types of user agent strings among the human outlier sessions. However, we can list various types of browsers (derived from user agent strings) which were utilized by human visitors that 'behave' like malicious crawlers, and those include: Mozilla Firefox, Internet Explorer, Safari, Google Chrome and Opera. Also there are a number of visitors that have utilized mobile platforms such as smart phones and tablet PCs.

The geographical distribution of human outlier sessions is presented in Table 4. Note that the information on geographic locations of the origin IP addresses was retrieved from the http://www.geobytes.com website. We have calculated that 25% of human outlier sessions are originating from the IP addresses located in Canada, 16% are from India, 10% are from United States, 4% from China, UK, Pakistan and Philippines, and 30% are originating from other mostly Asian countries. The sessions originating from Canadian IP addresses are in majority from Toronto area (about 17%) and about 60% of these are from the York University's workstations and on-campus computers. It is very possible that some of these outlier human sessions are actual human users with browsing characteristics similar to that of malicious crawlers. However, it is also possible that some of these are actual crawlers spoofing the user agent string of known browsers to avoid detection. It could also be the case that some of the identified geographical locations are, in fact, inaccurate since the malicious bots are known to frequently spoof the origin IP addresses.

Table 5
Mean, variance and significance of the differences test results on feature values between actual malicious sessions and human outlier sessions.

Features	Significant difference	Mean/variance actual malicious sessions	Mean/variance human outlier sessions
Click rate	Yes	0.059/0.029	0.74/1.98
Html to image ratio	Yes	29.63/66564	0.38/1.85
% of PDF documents	Yes	0.179/0.057	0.007/0.003
% of error requests	Yes	0.309/0.054	0.095/0.053
% of HEAD requests	No	0.009/0.0056	0.0017/0.00064
% of unassigned referrers	Yes	0.593/0.194	0.06/0.041
Number of bytes requested	Yes	4.87/1.61	5.52/1.21
Popularity Index	No	1.85/2.5	2.03/2.65
S.D. of page depth	Yes	1.08/0.34	0.5/0.4
% of sequential requests	Yes	0.441/0.064	0.663/0.097

5.3.2.2. (b) Significance of the difference test. As in the previous section, this group of outlier sessions has also been evaluated in terms of the significance of the difference test, this time relative to the actual malicious sessions. The results of the significance of the difference test (Eq. (3)), with 97.5% confidence, are shown in Table 5.



a) *GoogleBot* sessions hits

b) *MSNBot* sessions hits

c) *YahooBot* sessions hits

Fig. 4. *GoogleBot*, *MSNBot* and *YahooBot* crawler session hits per neuron visualized in a 2-dimensional SOM map. (a) *GoogleBot* sessions hits; (b) *MSNBot* sessions hits; (c) *YahooBot* sessions hits.

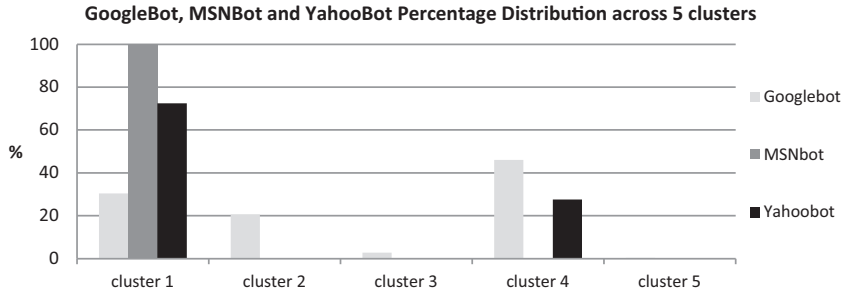


Fig. 5. *GoogleBot*, *MSNBot* and *YahooBot* crawler session percentage distribution across 5 clusters.

Table 6
Class distribution of well-behaved crawlers in the dataset.

	Number of sessions
Total # of GoogleBot sessions	326
Total # of MSNBot sessions	453
Total # of YahooBot sessions	69

Again, for each feature, we display the mean and variance values. Based on the presented results, the human sessions that 'behave' like malicious crawlers are shown to exhibit malicious-crawler-like browsing characteristics in terms of mean values of features 5 and 8. Put another way, the greatest similarity between human sessions that 'behave' like malicious crawlers and the actual malicious crawler sessions is in: (1) relatively high percentage of HEAD request, and (2) relatively small popularity of retrieved web pages. (Note, the maximum popularity index observed in our study was max(PPI) = 9.68.) Hence it is our recommendation that web administrators pay special attention and closely inspect sessions that satisfy these two requirements while at the same time carrying a valid user-agent string.

5.4. Well-behaved crawler analysis: Google, MSN, and Yahoo

In addition to performing in-depth evaluation of the outlier malicious and outlier human sessions, we have also taken a closer look at one particular subgroup of well-known and well-behaved crawlers. Namely, our goal was to investigate whether there are any significant similarities among the browsing styles of three popular commercial web crawlers: GoogleBot (owned by Google online search engine), MSNBot (owned by MSN online search engine) and YahooBot (owned by Yahoo online search engine).

The overall numbers of sessions associated with these three crawlers, and identified in our logs (i.e., data set), are shown in Table 6. In the first stage of the analysis, the given sessions are clustered using a 10-by-10 neuron SOM. The resultant SOM-maps are shown in Fig. 4. In particular, the maps in Fig. 4a–c show the neuron hits for sessions that were pre-labeled as belonging to GoogleBot, MSNBot, and YahooBot, respectively. In the second stage of the analysis, the same subgroup of sessions is clustered using Modified ART2. The respective results are plotted in Fig. 5, showing the percentage of each session type (GoogleBot, MSNBot, and YahooBot) per each of the five formed clusters. By inspecting Figs. 4 and 5, we derive the following conclusions:

- From Fig. 4, it is evident that sessions classified as belonging to GoogleBot, MSNBot and YahooBot are pretty well separated in the map. Namely, the sessions belonging GoogleBot end up firing neurons mostly in the right half of the map (and mostly in the lower right corner of the map) while sessions belonging to MSNBot end up firing neurons almost exclusively in the far left side of the map. The sessions belonging to YahooBot are mostly grouped around the center of the neuron map, though there is a noticeable overlap of neuron hits among sessions belonging to YahooBot and sessions belonging to the other two crawlers. This observation implies that YahooBot shares some crawling behavioral characteristics of the other crawlers, while GoogleBot and MSNBot exhibit unique crawling behaviors relative to each other.
- From Fig. 5, similar observation regarding the relative similarities and differences among GoogleBot, MSNBot and YahooBot sessions can be made. Namely, 72% of YahooBot sessions belong to cluster 1 together with MSNBot and GoogleBot sessions, while the remaining 28% of YahooBot sessions belong to cluster 4 together with GoogleBot sessions, thus implying a relatively high correlation between the browsing style of YahooBot and the other two crawlers. At the same time, only 30% of GoogleBot and MSNBot end up in the same joint cluster (cluster 1), with the remaining 70% of GoogleBot sessions being spread over clusters 2–5, i.e., clusters with no MSNBot membership. This suggests that: (1) there exist a rather high dissimilarity between the browsing styles of MSNBot and GoogleBot, and (2) GoogleBot browsing style is far more diverse than the browsing style of MSNBot (as well as YahooBot).

6. Conclusion and final remarks

The detection of malicious web crawlers is one of the most active research areas in the field of network security. In this paper, we approach the problem of malicious web-crawler detection and analysis through the use of unsupervised neural network learning.

The following important conclusions were derived from our study:

In general, there exists a reasonably good separation between malicious and non-malicious web users in terms of their browsing behavior. There also seems to be a pretty good separation in crawling behavior of the three most well-known crawlers, namely, GoogleBot, MSNBot and YahooBot. And, while mostly human visitors tend to follow rather similar browsing patterns, automated web crawlers (and in particular malicious web crawlers) exhibit a range of browsing strategies. Moreover, 52% of malicious web crawlers employ browsing strategies that are somewhat similar to those of regular human web-visitors. Our research shows that the majority of these malicious crawlers either do not set the user agent strings or simply spoof them in order to appear as regular human visitors. Clearly, with a higher level of sophistications, these crawlers could pose a serious challenge for current web-site security systems, especially those that perform simple screening of their visitors, e.g., by examining the value of user-agent string and/or looking for attempts to access the robots.txt file.

References

[1] C. Wilson, Botnets, Cybercrime, and Cyberterrorism: Vulnerabilities and Policy Issues for Congress, Foreign Affairs, Defense, and Trade Division, United States Governemnt, CRS Report for Congress, 2008.

[2] Prolexic Technologies, Evolving Botnet Capabilities – and What This Means for DDoS, White Paper, 2010.
[3] Y. Xie, S.-Z. Yu, Monitoring the application-layer DDoS attacks for popular websites, IEEE/ACM Transactions on Networking 17 (February (1)) (2009) 15–25.
[4] G. Oikonomou, J. Mirkovic, Modeling human behavior for defense against flash-crowd attacks, in: Proceedings of IEEE International Conference on Communications, Dresden, Germany, 2009, pp. 1–6.
[5] P. Hayati, V. Potdar, K. Chai, A. Talevski, Web spambot detection based on web navigation behaviour, in: International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 797–803.
[6] C. Bomhardt, W. Gaul, L. Schmidt-Thieme, Web robot detection – preprocessing web logfiles for robot detection, in: Proc. SISCLADAG, Bologna, Italy, 2005.
[7] K. Park, V. Pai, K. Lee, S. Calo, Securing web service by automatic robot detection, in: Proceedings of the annual conference on USENIX'06 Annual Technical Conference, Berkeley, CA, 2006, pp. 23–29.
[8] T. Kohonen, Self-Organizing Maps, 3rd ed., Springer-Verlag, Berlin Heidelberg, New York, 2001.
[9] N. Vlajic, H.C. Card, Vector quantization of images using modified adaptive resonance algorithm for hierarchical clustering, IEEE Transactions on Neural Networks 12 (September (5)) (2001) 1147–1162.
[10] D. Stevanovic, A. An, N. Vlajic, Feature Evaluation for Web Crawler Detection with Data Mining, Elsevier Expert Systems with Applications Journal 39 (August (10)) (2012) 8707–8717.
[11] http://www.cs.waikato.ac.nz/ml/weka/
[12] P.N. Tan, V. Kumar, Discovery of web robot sessions based on their navigation patterns, Data Mining and Knowledge Discovery 6 (January (1)) (2002) 9–35.
[13] A. Stassopoulou, M.D. Dikaiakos, Web robot detection: a probabilistic reasoning approach, Computer Networks: The International Journal of Computer and Telecommunications Networking 53 (February (3)) (2009) 265–278.
[14] Y. Hiltunen, M. Lappalainen, Automated personalization of internet users using self-organizing maps, in: IDEAL, Manchester, UK, 2002, pp. 31–34.
[15] D. Petrilis, C. Halatsis, Two-level clustering of web sites using self-organizing maps, Neural Process Letters 27 (February (1)) (2008) 85–95.
[16] J. Martín-Guerrero, E. Soria-Olivas, P.J.G. Lisboa, A. Palomares, E. Balaguer-Ballester, User profiling from citizen web portal accesses using the adaptive resonance theory neural network, in: IADIS, San Sabastian, Spain, 2006, pp. 334–337.
[17] http://en.wikipedia.org/wiki/Self-organizing_map
[18] http://en.wikipedia.org/wiki/Adaptive_resonance_theory
[19] J. Leea, S. Chab, D. Leec, H. Leec, Classification of web robots: an empirical study based on over one billion requests, Computers & Security 28 (November (8)) (2009) 795–802.
[20] D. Doran, S.S. Gokhale, Web robot detection techniques: overview and limitations, Data Mining and Knowledge Discovery 22 (1-2) (2010) 183–210.
[21] User-Agents.org [online], http://www.user-agents.org (2011 August).
[22] Bots vs. Browsers [online], http://www.botsvsbrowsers.com/ (2011 August).
[23] R. Wonnacott, T. Wonnacott, Introductory Statistics, 4th ed., John Wiley and Sons, USA, 1996.
[24] G.A. Carpenter, S. Grossberg, Adaptive resonance theory, in: M.A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, USA, 1998, pp. 79–82.