Expert Systems with Applications 39 (2012) 8707–8717

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Feature evaluation for web crawler detection with data mining techniques

Dusan Stevanovic *, Aijun An, Natalija Vlajic

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

ARTICLE INFO

Keywords:
Web crawler detection
Web server access logs
Data mining
Classification
DDoS
WEKA

ABSTRACT

Distributed Denial of Service (DDoS) is 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 effect of applying seven well-established data mining classification algorithms on static web server access logs in order to: (1) classify user sessions as belonging to either automated web crawlers or human visitors and (2) identify which of the automated web crawlers sessions exhibit 'malicious' behavior and are potentially participants in a DDoS attack. The classification performance is evaluated in terms of classification accuracy, recall, precision and F_1 score. Seven out of nine vector (i.e. web-session) features employed in our work are borrowed from earlier studies on classification of user sessions as belonging to web crawlers. However, we also introduce two novel web-session features: the consecutive sequential request ratio and standard deviation of page request depth. The effectiveness of the new features is evaluated in terms of the information gain and gain ratio metrics. The experimental results demonstrate the potential of the new features to improve the accuracy of data mining classifiers in identifying malicious and well-behaved web crawler sessions.

1. Introduction

Today, the world is highly dependent on the Internet, the main infrastructure of the global information society. Therefore, the availability of Internet is very critical for the economic growth of the society. 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. These services are now being actively replaced by cheaper and more efficient Internet-based applications. However, the inherent vulnerabilities of the Internet architecture provide opportunities for various attacks on the security of Internet-based applications. For example, distributed denial-of-service (DoS) is a type of security attack that poses an immense threat to the availability of any Internet-based service and application. The DoS effect is achieved 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 (see Fig. 1). In general, single-source DoS attacks can be easily prevented by locating and disabling the source of the malicious traffic. However, distributed DoS (DDoS) attacks launched from hundreds to tens of thousands of compromised zombies can present a much more complex challenge. Namely, unlike in the single-source DoS attack scenarios, the problem of locating the malicious hosts responsible for a DDoS attack becomes extremely difficult due to the sheer number of hosts participating in the attack. Furthermore, the larger collection of malicious hosts can generate enormous amount of traffic towards the victim. The result is a substantial loss of service and revenue for businesses under attack. According to the United States' Department of Defence report from 2008 presented in Wilson et al. (2008), cyber attacks from individuals and countries targeting economic, political, and military organizations may increase in the future and cost billions of dollars.

Now, attackers launching the traditional DDoS attacks by employing illegal Network Layer packets can be easily detected (but not easily stopped) by the signature detections systems such as intrusion detection systems. However, an emerging (and increasingly more prevalent) set of DDoS attacks known as Application Layer or Layer-7 attacks are shown to be particularly challenging to detect. The traditional network measurement systems often fail to identify the presence of Layer-7 DDoS attacks. The reason for this is that in an Application Layer attack, the attacker utilizes a legitimate network session. More specifically, the attacker utilizes a web crawler[1] program that performs a clever semi-random walk of the web site links, intended to resemble the web site traversal of an

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

[1] Crawlers are programs that traverse the Internet autonomously, starting from a *seed* list of web pages and recursively visit documents accessible from that list. Crawlers are also referred to as robots, 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 crawl the Web to compare prices and products sold by different e-commerce sites.

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.210

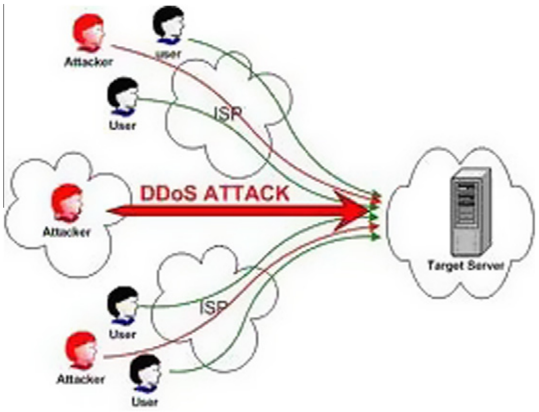

Fig. 1. Application layer denial of service attack.

actual human user. Since such attack signatures look very much like legitimate traffic, it is difficult to construct an effective metric to detect and defend against the Layer-7 attacks.

Numerous studies have been published on the topic of Layer-7 DDoS attacks. Given that the key challenge of Layer-7 DDoS attacks is their close similarity to the patterns of legitimate user traffic; researchers studying Layer-7 defence mechanism are mostly interested in devising effective techniques of attack detection. More specifically, the research works in this field are categorized into two main groups: (1) detection of application-layer DDoS attacks during a *flash crowd* event based on aggregate-traffic analysis (Oikonomou & Mirkovic, 2009; Xie & Yu, 2009) and (2) differentiation between wellbehaved and malicious web crawlers based on web-log analysis (Bomhardt, Gaul, & Schmidt-Thieme, 2005; Hayati, Potdar, Chai, & Talevski, 2010; Park, Pai, Lee, & Calo, 2006). (A more detailed overview of the works from the latter group is provided in Section 2.)

In this study, we pursue the line of research of the second group of works further, through two sets of experiments. In particular, the goal of the first set of experiments is to: (1) examine the effectiveness of seven selected classification algorithms in detecting the presence of (i.e. distinguish between) known well-behaved web crawlers and human visitors and (2) evaluate the potential of two newly proposed web-session features to improve the classification accuracy of the examined algorithms. The goal of the second experiment is to: (1) examine the effectiveness of seven classification algorithms in distinguishing between four visitor groups to a web site (malicious web crawlers, well-behaved web crawlers, human visitors and unknown visitors (either human or robot)) and (2) evaluate the potential of two newly proposed web-session features to improve the classification accuracy of the examined algorithms in this particular case. The datasets used in the experiments are generated by pre-processing web server access log files. The implementations of classification algorithms are provided by WEKA data mining software (WEKA, 2010).

The novelty of our research is twofold. Firstly, to the best of our knowledge, this is the first study that looks into the detection of the so-called malicious web crawlers, i.e. crawlers used to conduct Layer-7 attacks, and ways of distinguishing them from well-behaved web robots (such as Googlebot and MSNbot among others). Secondly, in addition to employing traditional web-session features in our classification, we also introduce two new features and prove that the utilization of these features can improve the classification accuracy of the examined algorithms.

The paper is organized as follows: In Section 2, we discuss previous works on web crawler detection. In Section 3, we discuss the advantages of utilizing supervised learning for the purpose of web visitor detection over using a simple rule-based web-log analyzer. In Section 4, we present an overview of our web-log analyzer and the process of dataset generation and labelling. In Section 5, we outline the design of the experiments and the performance metrics that were utilized. In Section 6, we present and discuss the results obtained from the classification study. In Section 7, we conclude the paper with our final remarks.

2. Related work

In over the last decade, there have been numerous studies that have tried to classify web robots from web server access logs. One of the first studies on classification of web robots using data mining classification techniques is presented in Tan and Kumar (2002). In this study, the authors attempt to discover web robot sessions by utilizing feature vectors derived from a number of different properties of recorded user sessions. In the first step, the authors propose a new approach to extract sessions from log data. They argue that the standard approach based on grouping web log entries according to their IP address and user-agent fields may not work well since an IP/user-agent pair may contain more than one session (for example, sessions created by web users that share the same proxy server). Next, authors derive 25 different properties of each session by breaking down the sessions into episodes, where an episode corresponds to a request for an HTML file. Among 25 different properties or features, authors identify three features that, in their belief, most distinctly represent sessions likely to be robots, and therefore can be used for the purposes of class labelling. These three features are: (1) checking if robots.txt (file that lists pages that may be accessed by the robots) was accessed, (2) the percentage of page requests made with the HTTP method of type HEAD and (3) percentage of requests made with an unassigned referrer field. These features most distinctly represent sessions likely to be robots since normally a human user would not request robots.txt, send a large number of HEAD requests, or send requests with unassigned referrer fields. As a result of the initial class labelling, the observed user sessions are partitioned into groups of known robots, known browsers, possible robots, and possible browsers. Finally, the technique adopts the C4.5 decision tree algorithm over the labelled human and robot sessions using all of the 25 derived navigational features. This classification model when applied to a dataset suggests that robots can be detected with more than 90% accuracy after only four requests.

In addition to C4.5, other data mining techniques have also been used for the purposes of log-session classification. In Bomhardt et al. (2005) and Stassopoulou and Dikaiakos (2009), for example, authors utilize Bayesian classification and neural networks respectively, to detect web robot presence in web server access log files. Many of the features used in Bomhardt et al. (2005) and Stassopoulou and Dikaiakos (2009) overlap with those from (Tan & Kumar, 2002), indicating an emerging consensus on what metrics should be used to characterize web robot traffic. Examples of works that utilize other (unrelated to data-mining) methods of robot identification are (Wei-Zhou & Shun-Zhenga, 2006) (use Markov Chain modelling), (Ahn, Blum, Langford, & Hopper, 2003) (use Turing tests), and (Lin, Quan, & Wu, 2008) (use aggregate traffic analysis). The malicious crawler detection has been addressed in studies in (Lin, 2009; Hou, Chang, Chen, Laih, & Chen, 2010).

3. Supervised learning versus log parser

Many of the early systems for classification of web-site visitors were based on simple rule-based logic. Namely, for any given weblog file, a rule-based classification system would first perform text pre-processing in order to identify (i.e., extract) individual user sessions. Subsequently, by focusing on one or a few particular features, the system would derive a numerical (i.e. vector) representation of each identified session. Finally, by applying a set of pre-established hard-threshold rules to the derived session representations, the system would decide which web-site visitor group each session should be placed in. In addition to the classification rules, the number and type of visitor groups also had to be manually defined, by relying on the judgment and experience of the administering staff. Clearly, the performance of such systems was greatly dependant on the accuracy of the established classification rules and visitor groups. That is, insufficient experience of the administering staff was likely to produce questionable clustering results.

Newer systems for web-log analysis employ more sophisticated approaches to visitor classification – in some cases involving the use of machine learning. Several of these approaches, as previously reported in the research literature, have been outlined in Section 2.

In this study, we look at the utilization of supervised learning (i.e. supervised data-mining algorithms) for the purpose of web-visitor detection. In the machine learning terminology, supervised learning refers to a group of algorithms that attempt to create a classification model from a pre-labelled (training) data samples, and then subsequently allow the use of this model for classification of new previously unseen data. From the perspective of web-visitor classification, the use of supervised learning implies an intelligent and automated formation of classification rules without direct human intervention – unlike what has been seen in rule-based systems. Moreover, classification rules formed through the process of supervised learning are more likely to be 'in tune' with the actual nature of the presented (training) data and, consequently, produce more accurate classification results on new data samples, e.g., previously unseen types of visitor sessions.

4. Dataset preparation

Supervised data-mining algorithms require pre-labelled training samples in order to learn (i.e. build) a classification model for a particular dataset. In this section we give a brief overview of our log analyser that has been used to generate a workable dataset – comprising both training and testing data samples – from any given web-log file. The operation of the log analyser is carried out in three stages: (1) session identification, (2) feature extraction for each identified session, and (3) session labelling (see Fig. 2).

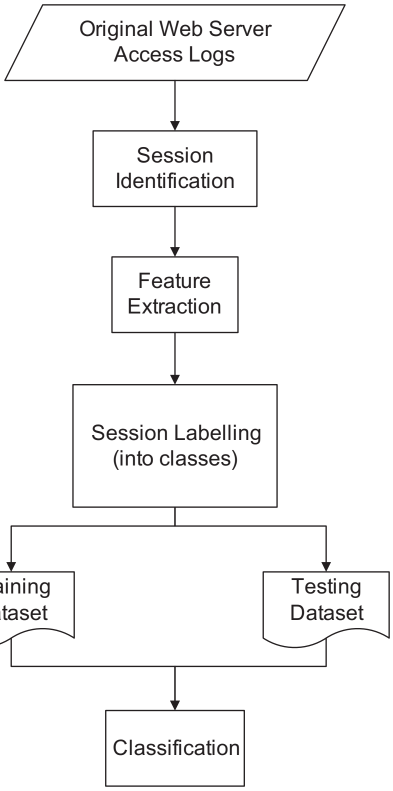

Fig. 2. Web server access log pre-processing.

4.1. Session identification

Session identification is the task of dividing a server access log into individual web sessions. According to Liu and Keselj (2007), a web session is a group of activities performed by one individual user from the moment he enters a web site to the moment he leaves it. Session identification is typically performed first by grouping all HTTP requests that originate from the same IP address and the same user-agent, and second by applying a timeout approach to break this grouping into different sub-groups, so that the time-lapse between two consecutive sub-groups is longer than a pre-defined threshold. The key challenge of this method is to determine proper threshold-value, as different Web users exhibit different navigation behaviours. In the majority of web-related literature, 30-min period has been used as the most appropriate maximum session length (see Stassopoulou & Dikaiakos, 2009; Tan & Kumar, 2002). Hence, our log analyser employs the same 30-min threshold to distinguish between different sessions launched by the same user.

4.2. Features

A typical web server access log file includes the information such as the IP address/host name of the site visitor, the page requested, 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 requested page. These fields may be used to identify specific features that characterize a particular user session. From previous web crawler classification studies, namely (Bomhardt et al., 2005; Stassopoulou & Dikaiakos, 2009; Tan & Kumar, 2002; Yu, O, Zhang, & Zhang, 2005), we have adopted seven different features that are shown to be useful in distinguishing between browsing patterns of web robots and humans. These features are enlisted below. (Note that in the rest of the paper we will refer to these features based on their numeric ID shown here):

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. 'Robots.txt' file request – a *nominal* attribute with values of either 1 or 0, indicating whether 'robots.txt' file was or was not requested by a user during a session, respectively. Web administrators, through the Robots Exclusion Protocol, use a special-format file called *robots.txt* to indicate to visiting robots which parts of their sites should not be visited by the robot. For example, when visiting a Web-site, say http://www.cse.yorku.ca, a robot should first check for http://www.cse.yorku.ca/robots.txt in order to learn about possible access limitations. It is unlikely that any human would check for this file, since there is no external or internal hyperlink leading to this file, nor are (most) users aware of its existence.

   As mentioned earlier, features 1–7 have been used in the past for distinguishing between human- and robot-initiating sessions. However, based on the recommendations and discussion presented in Doran and Gokhale (2010), we have derived two additional and novel features for the purpose of web robot classification:
8. 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'.
9. 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.

In Doran and Gokhale (2010), authors argue that analytical robot detection techniques must be based on fundamental distinctions between the robot and human traffic across server domains and in the face of evolving robot traffic. We argue that the features 8 and 9, which to the best of our knowledge have not been used in the previous research on web robot detection, have an excellent chance in separating human and robotic users in server access log sessions.

The importance of features 8 and 9 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 contrast, robots are neither expected to have such complex navigational patterns, nor would they be restricted by the link structure of the web site. Namely, since crawlers, like Google, browse systematically through an entire web-domain, i.e. they access files at various depths in the file hierarchy, their standard deviation should be large. On the other hand, humans tend to concentrate on one particular type of information, typically stored over a few files in a single directory. For the above reasons, the standard deviation of requested pages' depths, i.e. feature 8, should be high for web robot sessions and low for sessions belonging to human users. Note that feature 8 will be effective at distinguishing crawlers from human visitors only when applied on log files generated from web sites with large number of distinct web pages such as a University department website.

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 because 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 if they have the capacity to do so. For the above reasons, the number of consecutive sequential HTTP requests should be high in human user sessions and low in web robot sessions.

Finally, note, since we are investigating the behaviour as evident from the click-stream of a user-agent, it is fair to assume that any session with less than 5 requests is too short to enable labelling, even by manual inspection. We are therefore ignoring sessions that are too small (i.e. with less than 5 requests in total).

4.3. Dataset labelling

After the log analyzer parses the log file and extracts the individual visitor sessions, each session (i.e. the respective feature vector) is labelled as belonging to a particular class. Subsequently, 70% of the feature vectors are placed in the training, and 30% of the feature vector into the testing dataset. The class labels in the training dataset are to serve as the learning model for the classification algorithms, while the class labels in the testing dataset are to be used as the references against which the classification accuracy of the classification algorithms is measured.

In this study, we perform two types of classifications/experiments:

(1) Experiment #1: Classification of human sessions and well-behaved web crawler sessions. For this experiment, the log analyzer generates a dataset in which human sessions are labeled with value 0 and sessions of well-behaved web crawler are labeled with value 1. Note that in this experiment, sessions classified as belonging to malicious web crawlers or unknown visitors are removed from the dataset.

(2) Experiment #2: Classification of human and well-behaved crawler sessions as one group, and sessions belonging to malicious web crawler and unknown visitors as another. In this experiment, the log analyzer generates a dataset in which human sessions or sessions of known well-behaved web crawlers are class labeled with value 0 and sessions of known malicious web crawlers or unknown visitors are labeled with value 1.

In the first experiment we would like to examine whether human users and well-behaved crawlers can be separated by the classification algorithms. This classification experiment is similar to what has been done in the previous works (Doran & Gokhale, 2010; Tan & Kumar, 2002). Namely, we attempt to distinguish crawlers from human visitors of the site. However, our work is unique since we perform the classification with additional novel features and seven well-established data mining classifiers.

In the second experiment we would like to examine whether data mining classifiers can be applied to separate human users and well-behaved crawlers from known malicious crawlers and unknown visitors. The motivation for this experiment is to investigate whether the browsing characteristics (as derived by the 9 features) of malicious crawlers and unknown visitors are sufficiently different from browsing characteristics of human visitors and well-behaved crawlers to enable automatic classification. To the best of our knowledge, this is also a unique experiment.

The dataset labelling for two experiments is described in the following two sections.

4.3.1. Dataset labelling in Experiment 1

The log analyzer maintains a table of user agent fields of all known (malicious or well-behaved) web crawlers. This table is built using the data found on web sites (Bots vs. Browsers, 2011; User-Agents.org, 2011). (The web sites also maintain the list of various browser user agent strings that can be used to identify human visitors to the site as well.) Given the information contained in this table, the dataset labelling employed in the first experiment is performed as follows:

1. If the user agent field of a session matches the entry in the table of a known well-behaved crawler, the respective feature vector is labelled as such (the vector's class label is set to 1).
2. If the user agent string matches the user agent string of a browser, the respective feature vector is labelled as belonging to a human visitor (the vector's class label is set to 0).
3. Otherwise, if neither of the above is satisfied, we ignore the session since it belongs either to the malicious crawler or unknown visitor.

4.3.2. Dataset labelling in Experiment 2

In the second experiment, we again utilize the table of known user agent fields in order to perform the dataset labelling. The labelling is performed as follows:

1. If the user agent string of a session matches the user agent string of a known browser or of a known well-behaved crawler, the respective feature vector is labelled with class label 0.
2. If the user agent string matches the user agent string of a known malicious crawler, or is not enlisted in the table of known user agents, the respective feature vector is labelled with class label 1.

5. Experimental design

In the previous Section, we have described the process of dataset preparation (including session identification, feature extraction and vector labelling) as performed by our log analyzer. Assuming the correctness of the labelling process, the main goal of our work is to examine the classification accuracy of seven selected supervised learning algorithms when applied to the prepared dataset, as well as to evaluate the effectiveness of utilized web-session features in improving the algorithms' classification accuracy. In this section, we outline the details of the experimental study that followed the dataset preparation process.

5.1. Experimental setup

In both experiments we have conducted two types of tests: one test involving only features 1–7 and the other involving all 9 features, as discussed in Section 4. Subsequently, the results of the two tests are compared in order to examine whether features 8 and 9 can improve the accuracy rate of the classification algorithms.

5.2. Web server access logs

The data sets are constructed by pre-processing web server access log files provided by York CSE department. The log file stores detailed information about user web-based access into the domain www.cse.yorku.ca during a 4-week interval – between mid December 2010 and mid January 2011. There are a total of about 3 million log entries in the file. Tables 1 and 2 list the number of sessions and class label distributions generated by the log analyzer for experiments 1 and 2, respectively.

A typical entry in the cse.yorku.ca server access log file resembles the following line of data:

> *122.248.163.1 - - [09/Jan/2011:04:37:38 -0500] "GET /course_archive/2008-09/W/3421/test/testTwoPrep.html HTTP/1.1" 200 5645 Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)*

Each file entry contains information in the following order from left to right: IP address of the source of the request (122.248.163.1), the timestamp of the request (09/Jan/2011:04:37:38 -0500), the HTTP method (GET), the file on the server that was requested (/course_archive/2008-09/W/3421/test/testTwoPrep.html), the response code from the server (200), the size of the data retrieved from the server (5645 bytes) and user agent field (Mozilla/5.0 compatible; Googlebot/2.1; +http://www.google.com/bot.html). The information provided in individual entries is employed by the log analyzer in all three stages of dataset preparation: session identification, feature-vector extraction, and dataset labelling.

5.3. Classification algorithms

The detection of web crawlers is evaluated with the following seven classifiers: C4.5 (Quinlan, 1993), RIPPER (Cohen, 1995), Naïve Bayesian, Bayesian Network, *k*-Nearest Neighbour (with *k* = 1), LibSVM and Neural Networks (Multilayer Perceptron). (More on the last five data mining classification algorithms can be found in Han & Kamber (2006)). The implementation of each

Table 1
Class distribution in training and testing datasets used in Experiment #1.

	Training examples	Testing examples
Total number of sessions	96,845	36,789
Total # of session with class label = 0	94,723	35,933
Total # of session with class label = 1	2122	856

Table 2
Class distribution in training and testing datasets used in Experiment #2.

	Training examples	Testing examples
Total number of sessions	99,427	37,714
Total # of session with class label = 0	96,845	36,789
Total # of session with class label = 1	2582	925

algorithm is provided in the WEKA software package. For all classifiers, the default set of parameters as specified in WEKA are used. Each classifier is first trained on the training dataset, and then tested on another supplementary dataset. In the testing phase, the classification results generated by the trained classifiers are compared against the test-data's respective session labels as they have originally been derived by the log analyzer (see Section 4.3).

5.4. Dataset up-sampling

A simple evaluation of imbalanced datasets based on accuracy, i.e. the percentage of correct classifications, can be misleading. To illustrate this, assume a dataset with 100 cases out of which 90 cases belong to the majority class and 10 cases belong to the minority class. Then a classifier that classifies every case as a majority class will have 90% accuracy, even though it failed to detect every single target of the minority class.

It is evident from Tables 1 and 2 that our original datasets suffered from serous class imbalance. In order to overcome this problem, and be able to conduct a more meaningful performance evaluation, we have applied the process of dataset up-sampling[2]. In particular, for each classification algorithm various up-sampling ratios were tried, and one that produced the best classification accuracy was ultimately used, i.e. kept.

5.4.1. Recall, precision and F_1 score

In order to test the effectiveness of our classifiers, we have adopted the following three metrics: recall, precision, and the F_1-score (Stassopoulou & Dikaiakos, 2009). The exact expressions for the three metrics, as used in Experiment 1 and 2, are presented in (1)–(5). Note that in both experiments, we evaluate the precision and recall scores for the underrepresented class only, i.e., Class 1. (Recall from Sections 4.3.1 and 4.3.2 that in Experiment 1 Class 1 comprises well-behaved crawlers, and in Experiment 2 it comprises malicious crawlers and unknown visitors.) This is justified by the fact that the main objective of our work is to be able to identify automated web-crawler visitors to a web-site, and in particular web-crawlers that exhibit malicious behaviour and/or intent. However, we also calculate the precision (accuracy of classification) for both classes as well (presented in (6) and (7) as utilized for experiments 1 and 2, respectively).

It is also worth nothing from (5) that F_1 score summarizes the first two metrics into a single value, in a way that both metrics are given equal importance. Namely, the F_1-score penalizes a classifier that gives high recall but sacrifices precision and vice versa. For example, a classifier that classifies all examples as positive has perfect recall but very poor precision. Recall and precision should therefore be close to each other, otherwise the F_1-score yields a value closer to the smaller of the two. The definition of these metrics is given below:

$$\text{Experiment 1}: \quad \text{Precision(class 1 only)} = \frac{\#\text{ of well} - \text{behaved crawler sessions correctly classified}}{\#\text{ of predicted well} - \text{behaved crawler sessions}} \tag{1}$$

$$\text{Experiment 2}: \quad \text{Precision(class 1 only)} = \frac{\#\text{ of (malicious or unknown) sessions correctly classified}}{\#\text{ of predicted (malicious or unknown) crawler sessions}} \tag{2}$$

$$\text{Experiment 1}: \quad \text{Recall (class 1 only)} = \frac{\#\text{ of well} - \text{behaved crawler sessions correctly classified}}{\#\text{ of actual well} - \text{behaved crawler sessions}} \tag{3}$$

$$\text{Experiment 2}: \quad \text{Recall (class 1 only)} = \frac{\#\text{ of (malicious or unknown) sessions correctly classified}}{\#\text{ of actual (malicious or unknown) sessions}} \tag{4}$$

$$F_1 = \frac{2 * \text{Recall} * \text{precision}}{\text{Recall} + \text{precision}} \tag{5}$$

$$\text{Experiment 1}: \quad \text{Precision for both classes} = \frac{\#\text{ of (human or well} - \text{behaved) sessions correctly classified}}{\#\text{ of predicted (human or well} - \text{behaved) sessions}} \tag{6}$$

$$\text{Experiment 2}: \quad \text{Precision for both classes} = \frac{\#\text{ of (human, well} - \text{behaved, malicious or unknown) sessions correctly classified}}{\#\text{ of predicted (human, well} - \text{behaved, malicious or unknown) sessions}} \tag{7}$$

5.4.2. Information gain, gain ratio and significance of the difference test

In addition to ranking the classifiers, we also rank the most important dataset features by employing attribute selection methods such as information gain and gain ratio. The ranking provides the purity test of the two proposed features. Basically, a higher ranking, by either of the two metrics, implies that a feature is more valuable to a classifier in separating sessions into classes.

In addition, the effectiveness of the new attributes in classifying visitor's sessions is further evaluated by applying the significance of difference test or t-test. Namely, after we generate the classification results with all seven classifiers for both experiments, we separate the sessions in 2 groups, true negatives and true positives. The sessions are grouped into true negatives if both the log analyzer and classification algorithms label the session with class label 0. On the other hand, the sessions are grouped into true positives if both the log analyzer and classification algorithms label the session with class label 1.

Next, we calculate the means and variance of features 8 and 9 in both groups of sessions and perform the significance of the difference test with 97.5% confidence interval. The calculation of the significance of the difference test is based on the following formula:

$$t = \frac{|\text{mean}_1(f) - \text{mean}_2(f)|}{\sqrt{\frac{\text{Var}_1(f)}{n_1} + \frac{\text{Var}_2(f)}{n_2}}} \tag{8}$$

In the above equation mean_1 and mean_2 are means of the feature values in two groups, Var_1 and Var_2 are the variances of the feature values in two groups, and n_1 and n_2 are the number of elements in two groups. The degrees of freedom value used in the t-test is $n_1 + n_2 - 1$. The significance of the difference test is explained in greater detail in Wonnacott and Wonnacott (1996).

6. Classification results

In this Section we present and discuss the results of our two experiments. Namely, in Sections 6.1 and 6.2, we give a detailed

[2] Up-sampling is a data mining pre-processing technique that balances the class distribution in the dataset by duplicating the training examples belonging to the class with fewer samples. The amount of up-sampling can be controlled by the number of training examples that are duplicated.

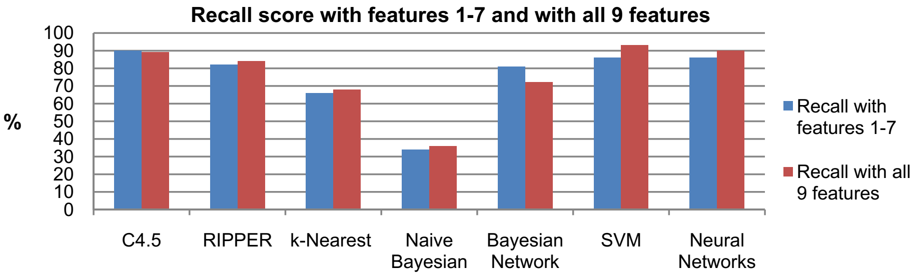

Fig. 3. Recall score for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

summary of the results of Experiments 1 and 2, respectively. In Section 6.3, we derive additional observation and conclusions from the presented results.

6.1. Experiment 1

In this section, we present the results derived in the first experiment. The motivation for this experiment was to test the classification accuracy of the seven data-mining algorithms, as well as to evaluate whether features 8 and 9 (see Section 4.2), can improve the accuracy in classifying sessions as either belonging to a human user or a well-behaved web crawler.

6.1.1. Recall, precision and F_1 score (class 1 only)

The comparisons of the recall, precision and F_1 scores for the datasets with features 1–7 and the dataset with all 9 features are shown in Figs. 3–5, respectively. It is evident from the presented graphs that the use of all 9 features generally improves both, the recall and precision scores, for all classifiers except for the Bayesian Network and C4.5 (only the recall statistic is slightly lower in the case of the C4.5 algorithm with all 9 features). Similarly, the use of 9 features results in improved F_1 score (between 0.5% up to nearly 4%) in six out of seven examined algorithms.

6.1.2. Classification precision for both classes (i.e. classification accuracy)

The Fig. 6 shows the comparison of the classification accuracy rate when the seven classification algorithms are trained on the datasets containing only features 1–7 and the dataset containing all 9 features. As expected, due to class imbalance, the classification accuracy is very high (at 95% or above) for all seven classification algorithms. However, as can be observed, there is a slight improvement in accuracy rate when all 9 features are used for all algorithms except the Bayesian Network which shows a slight decline in the accuracy rate.

6.1.3. Entropy-based feature ranking

Table 3 shows the ranking between all 9 features in terms of information gain and gain ratio metrics. As expected, feature 7 – a nominal value that indicates whether robots.txt file has been accessed during a session (see Section 4.2) – is at the top of the rankings for both metrics (has the highest gain ratio and second highest information gain). As explained in Section 4.2, well-behaved web crawlers are known to access the robots.txt file every time they visit a site while human visitors are not expected to request such a file. The percentage of unassigned referrers is another feature that defines whether a session belongs to a web crawler. Recall,

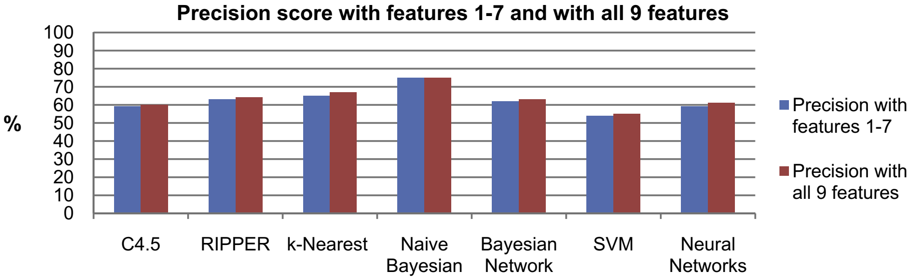

Fig. 4. Precision score for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

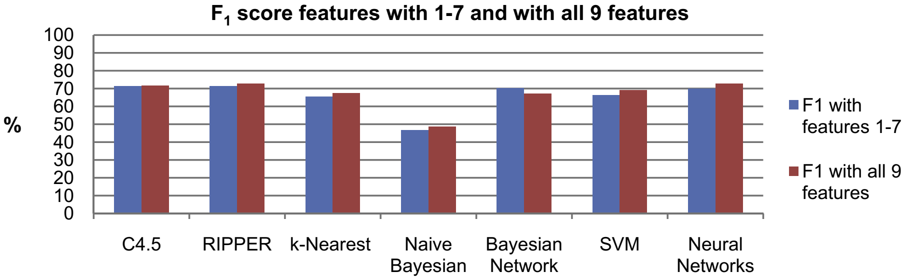

Fig. 5. F_1 score for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

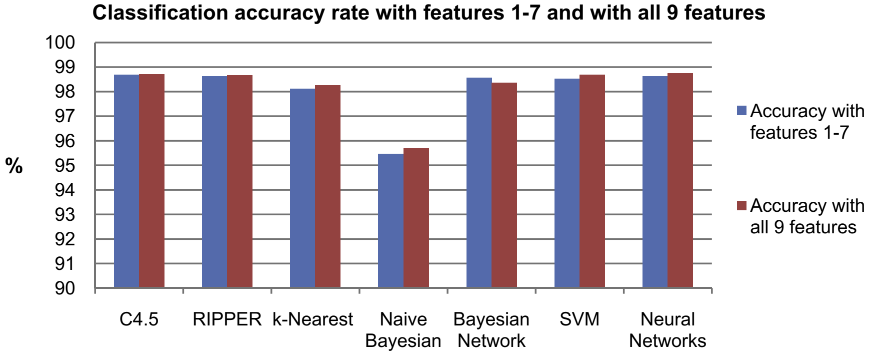

Fig. 6. Classification accuracy rate for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

Table 3
Attribute ranking in terms of information gain and gain ratio metrics (ordered top down from best to worst).

Information gain	Gain ratio
1. % of unassigned referrers	1. 'robots.txt' is requested
2. 'robots.txt' is requested	2. % of unassigned referrers
3. % of sequential HTTP requests	3. % of sequential requests
4. Click number	4. % of HEAD requests
5. Standard deviation of page depth	5. Click number
6. % of error requests	6. % of PDF documents
7. % of PDF documents	7. Standard deviation of page depth
8. HTML to Image ratio	8. % of error requests
9. % of HEAD requests	9. HTML to image ratio

Table 4
t-Scores for difference test on mean values of features 8 and 9 between true positive and true negative sessions.

Classifiers	Standard deviation of page depth (*t*-scores)	% of sequential requests (*t*-scores)
C4.5	7.76	8.00
RIPPER	7.12	7.15
k-Nearest neighbour	7.11	6.81
Naive Bayesian	6.20	5.99
Bayesian network	7.91	7.74
SVM	7.72	8.14
Neural network	7.78	8.24

the referrer parameter is typically only assigned by a browser of the user visiting the web site and is left blank if the visitor is a web crawler.

It is also interesting to note that the two new features introduced in this study are near the top of the rankings. The percentage of consecutive sequential requests is in the third position in both columns in the table. This implies that this attribute can be very helpful in determining whether the session belongs to a human user or a web crawler. Typically, large number of consecutive sequential requests (in the same directory of the web site) can only be attributed to a human user.

The standard deviation of the page depth also emerges as a fairly important feature – ranked 5th and 7th in the two respective columns. As explained in Section 4.2, web crawlers are generally expected to have higher standard deviation of requested page depth. Though, certain types of crawlers focus on specialized narrow searches and, as a result, have small standard deviation of page requests – something typically expected of a human visitor.

The significance of the differences of values between true positive and true negative sessions for features 8 and 9 can be confirmed by applying the significant difference of the mean test (Eq. (8) or the *t*-test) (Wonnacott & Wonnacott, 1996). As can be observed in Table 4, the mean values of features 8 and 9 are significantly different between true positive and true negative sessions in the case of all seven classifiers (as specified in Wonnacott & Wonnacott (1996), if *t*-score > 1.96, then the difference is significant with 97.5 % certainty/confidence).

6.2. Experiment 2

In this section, we present the results derived in the second experiment of our study. The goal of the experiment was to test the classification accuracy of the seven data-mining algorithms, as well as to evaluate whether features 8 and 9 can improve the accuracy in classifying sessions as belonging to malicious web crawlers and unknown visitors.

6.2.1. Recall, precision and F_1 score (class 1 only)

The comparisons of the recall, precision and F_1 scores for the dataset with features 1–7 and the dataset with all 9 features are shown in Figs. 7–9, respectively. It is evident from the presented results that for all but two classifiers (SVM and Neural Network algorithm) the use of all 9 features results in noticeably higher recall, precision, and F_1 scores. Note, though, that F_1 scores for the SVM and Neural Network algorithm are not decreased, but remain unchanged regardless of the number of features used.

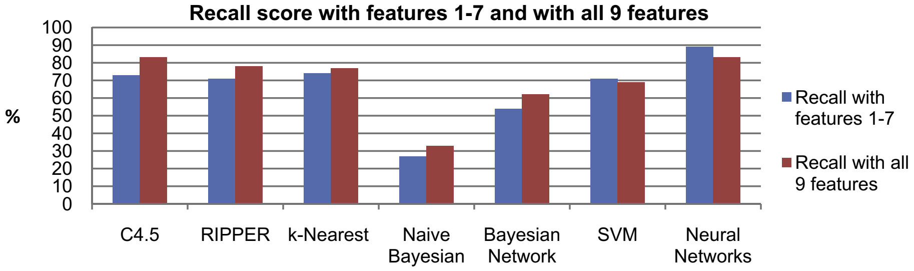

Fig. 7. Recall score for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

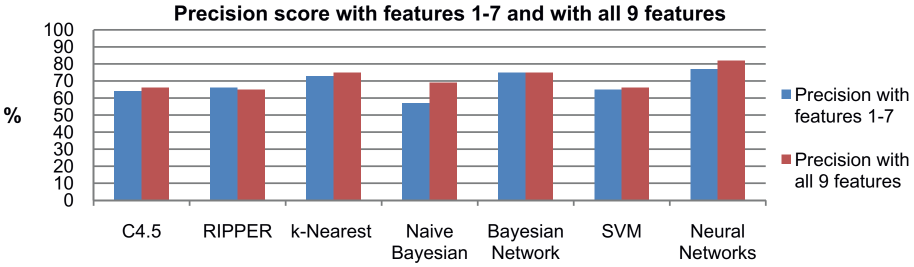

Fig. 8. Precision score for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

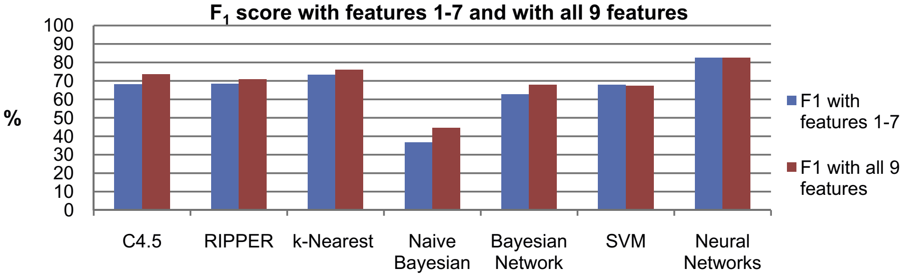

Fig. 9. F_1 score for various classifiers trained on the datasets that contain only features 1–7 and all 9 features.

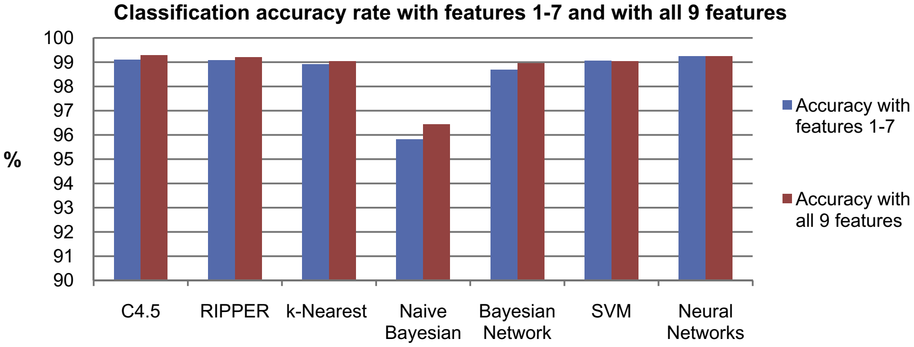

Fig. 10. Classification accuracy rate for various classifiers trained on the datasets containing only features 1–7 and all 9 features.

6.2.2. Classification precision for both classes (i.e. classification accuracy)

The Fig. 10 shows the comparison of the classification accuracy rate when the seven classification algorithms are trained on the dataset containing only features 1–7 and the dataset containing all 9 features. Again, as expected, due to class imbalance, the classification accuracy is very high (at 95% or above) for all seven classification algorithms. However, as can be observed, there is a slight improvement in accuracy rate when all 9 features are used in five out of seven examined algorithms.

6.2.3. Entropy-based attribute ranking

Lastly, Table 5 shows the ranking of 9 features in terms of information gain and gain ratio metrics. The ranking scores are rather similar to those from the first experiment shown in Table 3. Again,

Table 5
Attribute ranking in terms of information gain and gain ratio metrics (ordered top down from best to worst).

Information gain	Gain ratio
1. ‘robots.txt’ is requested	1. ‘robots.txt’ is requested
2. % of unassigned referrers	2. % of HEAD requests
3. Standard deviation of page depth	3. % of sequential HTTP requests
4. % of sequential HTTP Requests	4. % of unassigned Referrers
5. % of PDF documents	5. % of PDF documents
6. HTML to image ratio	6. Standard deviation of page depth
7. Click number	7. Click number
8. % of HEAD requests	8. % of error requests
9. % of error requests	9. HTML to image ratio

Table 6
t-Scores for difference test on mean values of features 8 and 9 between true positive and true negative sessions.

Classifiers	Standard deviation of page depth (*t*-scores)	% of sequential requests (*t*-scores)
C4.5	10.88	6.67
RIPPER	10.93	6.50
k-Nearest Neighbour	10.62	6.58
Naive Bayesian	9.16	6.25
Bayesian network	11.23	7.18
SVM	10.40	5.92
Neural network	10.83	6.53

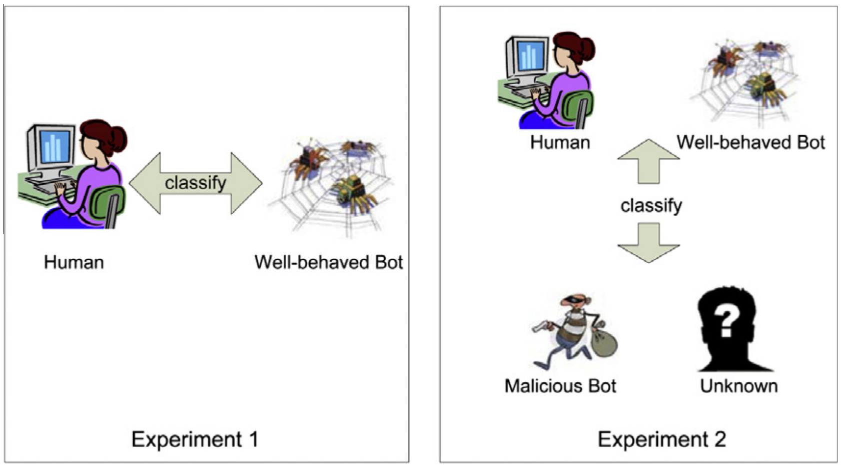

Fig. 11. Classification experiments.

as expected, the 'request for robots.txt file' and the 'rate of unassigned referrers in a session' are at the top of the rankings by both metrics.

The two features that we have proposed in this study are also near the top of the rankings. The 'consecutive sequential request' feature is ranked 4th according to the first and 3rd according to the second metric, while the 'standard deviation of the page depth of requests' is ranked 3rd according to the first and 6th according to the second metric.

The importance of the new features can be confirmed by applying the significant difference of the mean test on the features 8 and 9 (Eq. (8) or the t-test) (Wonnacott & Wonnacott, 1996). As can be observed in Table 6, the mean values of features 8 and 9 are significantly different between true positive and true negative sessions in the case of all seven classifiers (as specified in Wonnacott & Wonnacott (1996), if t-score > 1.96, then the difference is significant with 97.5 % certainty/confidence).

6.3. Discussion and observations

The results presented in the previous section show that most of the classifiers achieve fairly high recall and precision scores in both experiments. In particular, Neural Network, C4.5, RIPPER and k-Nearest Neighbour algorithms achieve the accuracy ratios close to 100%. Moreover, the F_1 scores for the underrepresented class (class 1) with Neural Network algorithm are 73% in Experiment 1 and 82% in Experiment 2. Finally, we can conclude that selected features are of high quality since in both experiments almost all sessions can be correctly placed into the defined classes.

The classification results of Experiment 1 are quite close to what we have expected to see. Namely, as discussed earlier, the characteristics of web site usage by well-behaved web crawlers is inherently different from the usage by human users in terms of the features examined in this study. Hence, it is (i.e. should be) fairly straightforward for a classification algorithms to differentiate between the two groups, i.e. their respective feature vectors.

The classification results of Experiment 2 demonstrate strong performance of classification algorithms when applied to the given problem. Namely, the classification task in Experiment 2 is more complex than the task in Experiment 1 because the classification algorithms are required to differentiate between four different types of sessions (see Fig. 11). Still, the Neural Network, C4.5, RIPPER and k-Nearest Neighbour algorithms can separate malicious robots and unknown visitors from well-behaved robots and human visitors with precision and recall scores above 70%. In fact, in experiment 2, the F_1 metric scores for the those four algorithms are higher than the F_1 metric scores for the same four algorithms in Experiment 1 (on average about 8% higher as you can observe by comparing Figs. 5 and 9).

7. Conclusion and final remarks

The detection of malicious web crawlers is one of the most active research areas in network security. In this paper, we study the problem of detecting known well-behaved web crawlers, known malicious web crawlers, unknown and human visitors to a web site using existing data mining classification algorithms.

The following three general conclusions were derived from our study:

- The classification accuracy of classification algorithms such as Neural Networks, C4.5, RIPPER and k-Nearest Neighbor algorithms is close to 100%. In the case of the Neural Network algorithm, the F_1 scores of the underrepresented class (class 1) are 73% and 82% in experiments 1 and 2, respectively.
- The two new features proposed – the consecutive sequential requests ratio and standard deviation of page request depths – are highly ranked among the other features used in the study by the information gain and gain ratio metrics. The new features are also explicitly shown to improve the classification accuracy, recall, precision and F_1 score of most evaluated algorithms, in both conducted experiments.
- As evident in our study, the browsing behaviours of web crawlers (both malicious and well-behaved) and human users are significantly different. Therefore, from the data mining perspective, their identification/classification is very much a feasible task. However, the identification/classification of crawlers that attempt to mimic human users will remain the most difficult future classification challenge. We believe that with customization of either C4.5 or RIPPER, the misclassification rates of known well-behaved and malicious web crawlers could be further reduced.

References

Ahn, L. v., Blum, M., Langford, J., & Hopper, N. (2003). CAPTCHA: Using hard AI problems for security. In *Proceedings of Eurocrypt* (pp. 294–311). Warsaw, Poland.

Bomhardt, C., Gaul, W., & Schmidt-Thieme, L. (2005). Web robot detection – Preprocessing web logfiles for robot detection. In *Proceedings of SISCLADAG*. Bologna, Italy.

Bots vs. Browsers. (2011). [Online]. http://www.botsvsbrowsers.com/.

Cohen, W.W. (1995). Fast effective rule induction. In *ICML 1995* (pp. 115–123).

Doran, D., & Gokhale, S. S. (2010). Web robot detection techniques: Overview and limitations. *Data Mining and Knowledge Discovery*, 1–28.

Han, J., & Kamber, M. (2006). In D. Cerra (Ed.). *Data mining: Concepts and techniques*. San Francisco, CA: Elsevier.

Hayati, P., Potdar, V., Chai, K., & Talevski, A. (2010). Web spambot detection based on web navigation behaviour. In *International conference on advanced information networking and applications* (pp. 797–803). Perth, Australia.

Hou, Y., Chang, Y., Chen, T., Laih, C., & Chen, C. (2010). Malicious web content detection by machine learning. *Expert Systems with Applications, 37*(1), 55–60.

Lin, J. (2009). Detection of cloaked web spam by using tag-based methods. *Expert Systems with Applications, 36*(4), 7493–7499.

Lin, X., Quan, L., & Wu, H., 2008. An automatic scheme to categorize user sessions in modern HTTP traffic. In *Proceedings of IEEE global telecommunications conference* (pp. 1–6). New Orleans, LA.

Liu, H., & Keselj, V. (2007). Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering, 61*(2), 304–330.

Oikonomou, G., & Mirkovic, J. 2009. Modeling human behavior for defense against flash-crowd attacks. In *Proceedings of IEEE international conference on communications* (pp. 1–6). Dresden, Germany.

Park, K., Pai, V., Lee, K., & Calo, S. (2006). Securing web service by automatic robot detection. In *Proceedings of the annual conference on USENIX '06 annual technical conference* (pp. 23–29). Berkeley, CA.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers.

Stassopoulou, A., & Dikaiakos, M. D. (2009). Web robot detection: A probabilistic reasoning approach. *Computer Networks: The International Journal of Computer and Telecommunications Networking, 53*(3), 265–278.

Tan, P. N., & Kumar, V. (2002). Discovery of web robot sessions based on their navigation patterns. *Data Mining and Knowledge Discovery, 6*(1), 9–35.

User-Agents.org. (2011). [Online]. http://www.user-agents.org.

Wei-Zhou, L., & Shun-Zhenga, Y. (2006). Web robot detection based on hidden Markov model. In *Proceedings of international conference on communications, circuits and systems* (pp. 1806–1810). Guilin, China.

WEKA (2010). [Online]. http://www.cs.waikato.ac.nz/ml/weka/.

Wilson, C. (2008). Botnets, cybercrime, and cyberterrorism: Vulnerabilities and policy issues for congress, foreign affairs, defense, and trade division, United States Governemnt, CRS Report for Congress.

Wonnacott, R., & Wonnacott, T. (1996). *Introductory statistics* (4th ed.). USA: John Wiley and Sons.

Xie, Y., & Yu, S.-Z. (2009). Monitoring the application-layer DDoS attacks for popular websites. *IEEE/ACM Transactions on Networking, 17*(1), 15–25.

Yu, J. X., O, Y., Zhang, C., & Zhang, S. (2005). Identifying interesting visitors through Web log classification. *Intelligent Systems, 20*(3), 55–59.