Detecting Web Crawlers from Web Server Access Logs with Data Mining Classifiers

Dusan Stevanovic
Department of Computer Science and Engineering, York University
4700 Keele Street
Toronto, ON, CA. M3J 1P3
+1 416 736 2100
dusan@cse.yorku.ca

ABSTRACT
Denial of Service (DoS) is one of the most damaging attacks on the Internet security today. Recently, malicious web crawlers have been used to execute automated DoS attacks on web sites across the WWW. In this study we examine whether seven well-established data mining classification algorithms may be employed to detect both well-behaved and malicious web crawlers from static web server access logs. We perform two experiments. In the first experiment we test the classification of known well-behaved web crawlers and in the second experiment we evaluate the classification of malicious web crawlers. The classification accuracy is evaluated in terms of recall, precision and F_1 score. The features used in our classification algorithms are mostly traditional features used to classify user sessions as belonging to web crawlers. However, we also introduce two novel features: the consecutive repeated request ratio and standard deviation of page request depth of requests. The experimental results demonstrate the potential of the two new features to improve the accuracy of web crawler classification by existing data mining classifiers.

Categories and Subject Descriptors
H.1.1 [**Computer Networks**]: Internet, Web Crawlers, Security
K.1.4 [**Data Mining**]: Classification Algorithms
G.2.3 [**Mining Software**]: WEKA
B.3.4 [**Programming Languages**]: Java and Unix Scripts

General Terms
Classification Algorithms, Denial of Service, Evaluation, Design, Web Crawler.

Keywords
Web Crawler Detection, Web Server Access Logs, Data Mining, Classification, WEKA.

1. INTRODUCTION
The phenomenal growth and success of Internet has changed the way traditional essential services such as banking, transportation, medicine, education and defense are operated. Now they are being actively replaced by cheaper and more efficient Internet-based applications. 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. However, the inherent vulnerabilities of the Internet architecture provide opportunities for various attacks on its security. Distributed denial-of-service (DDoS) is an ideal example of such an attack, which poses an immense threat to the availability of the Internet. United States' Department of Defence report from 2008, presented in [1], indicates that cyber attacks from individuals and countries targeting economic, political, and military organizations may increase in the future and cost billions of dollars.

This denial-of-service (DoS) effect is achieved by sending messages to the target (Internet host such as web site) that interfere with its operation, and make it hang, crash, reboot, or do useless work. In general, single-source DoS attacks can be easily prevented by locating the source of the malicious traffic and disabling it. However, DDoS attacks launched from thousands to millions of compromised hosts can present a much more complex challenge. As explained in [1], malicious software has reached unprecedented infection levels in 2009, with millions of computers compromised each month. Unlike in the DoS attack scenarios, the problem of locating the malicious hosts responsible for a DDoS attack becomes extremely difficult. Also, the attack itself is more vicious than its single-source counterpart since larger collection of malicious hosts can generate higher flood of traffic towards the victim. The result is substantial loss of service and revenue for businesses under attack.

In general, attackers launch the traditional DDoS attacks by employing illegal packets or network connections that can be easily detected (but not easily stopped) by the signature detections systems such as network firewalls. However, an emerging (and increasingly more prevalent) set of DDoS attacks known as Application Layer or Layer-7 attacks are extremely challenging to detect. The traditional network measurement systems often fail to identify the presence of Layer-7 DDoS attacks. The main reason is that in an application layer attack, the attacker utilizes a legitimate network session. For instance, HTML requests sent to a web server may be cleverly constructed to perform semi-random walks of web site links. Such an attack would resemble the web site traversal of an actual human user. Since the attack signature resembles legitimate traffic, it is difficult to construct an effective metric to detect and defend against the Layer-7 attacks.

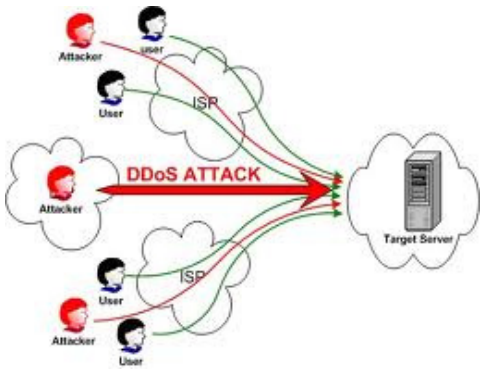

Figure 1: Application Layer Denial of Service Attack

Numerous studies have been published on the topic of application/layer 7 DDoS attacks. Given the fact that layer 7 DDoS attacks resemble the legitimate traffic; researchers studying layer 7 defense mechanism are mostly focused on attack detection. The research publications falls into two main groups: 1) DDoS attacks during a flash crowd event and 2) detection of 'malicious' web crawlers. In the first group, authors present techniques for detecting HTTP request floods during flash crowd events. In the second group of research works, authors attempt to classify web robots and to differentiate between well-behaved (such as search engine web crawlers) and malicious web crawlers (bots that automate the application-level DDoS attack, search for vulnerabilities and collect email addresses for spam).

In this study, the problem of malicious and non-malicious web crawler detection is examined. Namely, we performed two sets of experiments. In the first experiment, we attempt to detect the presence of known well-behaved web crawlers. In the second experiment, we attempt to detect the presence of malicious web crawlers among known well-behaved web crawlers and human visitors of a web site. The web crawlers are detected by classifying visitor's sessions with well-established data classification algorithms. The datasets used in the experiments are generated by preprocessing web server access log files. The implementations of classification algorithms are provided by WEKA data mining software [2].

The novelty of our research is twofold. Firstly, to the best of our knowledge, this is the first study that classifies web crawlers as malicious and known well-behaved web robots (such as Googlebot and MSNBot among others).Secondly, in addition to employing traditional features in our classification, we also introduce two new features and evaluate whether the utilization of these additional features can improve the classification accuracy rates.

The paper is organized as follows: In Section 2, we discuss previous works on web crawler detection. In Section 3, we present an overview of the web crawler classification by employing a simple log analyzer preprocessor. In Section 4, we outline the design of the experiments and the performance metrics utilized. In Section 5, we present and discuss the results obtained from the classification study. In Section 6, we conclude the paper with our final remarks and recommendations.

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 that attempts to classify web robots using data mining classification techniques is presented in [3]. The authors attempt to discover web robot sessions by utilizing a feature vector of the properties of Web sessions. In the first step, they 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). Therefore, to determine what session a log entry *l* belongs to, each active session is scanned to check the time difference between *l* and the current session, along with some unspecified session contiguity conditions. If this time difference exceeds a threshold or the conditions are not met, then a new session is generated starting with log entry *l*. Before scanning, all the active sessions are divided into four groups depending on whether the user-agent and IP address fields match those found in *l*. The first group consists of sessions where both user-agent and IP addresses match, followed by the two session groups with one matching field, and finally the group with no matching fields.

They then derive twenty-five different properties of each session by breaking down the sessions into episodes, where an episode corresponds to a request for an HTML file. These include checking if *robots.txt* (file that lists pages that may be accessed by the robots) was accessed, the percentage of page requests made with the HTTP method of type HEAD, and percentage of requests made with an unassigned referrer field. These features are used since they most distinctly represent sessions likely to be robots, assuming that normally a human user would not request *robots.txt*, send a large number of HEAD requests, or send requests with unassigned referrer fields.

From this initial class labelling, the observed user-agent fields are partitioned into groups of known robots, known browsers, possible robots, and possible browsers in the following manner. If a derived session *s* contains a request for *robots.txt*, the session is declared to be a robot. Otherwise, the user-agent fields of the requests in the session are considered. If *s* only ever has requests from one user agent, and the user agent is a known robot or a possible robot, then *s* is labelled as a robot. Otherwise, it is labelled as a human. If *s* has requests from multiple user agents, however, the session is labelled as a robot only if there are no sessions that are known browsers or possible browsers or if the session contains requests that all use the HEAD http method or requests that all have unassigned referrer fields.

Finally the technique adopts the C4.5 decision tree algorithm over the labelled human and robot sessions using all of the twenty-five derived navigational features. Their objective is to develop a good model to predict web robot sessions based only on access features and to detect robot traffic as early as possible during a robot's visit to the site. This classification model when applied to a data set suggests that robots can be detected with more than 90% accuracy after only four requests.

Other techniques have also been proposed that utilize similar classification methods described as can be found in [3]. In [4], authors utilize neural networks to detect web crawlers and compare their results to a decision tree technique. In [5], authors utilize Bayesian classification to detect web robot presence in web server access log files. Many of the features used in the three studies overlap indicating an emerging consensus on what metrics should be used to characterize web robot traffic.

However, not all techniques utilize data mining methods to detect the presence of web robots. It is important to list alternative methods that utilize Markov Chain modeling, presented in [6], Turing tests, presented in [7], and traffic characteristics, presented in [8], to uncover web robots.

3. WEB CRAWLER CLASSIFICATION

Crawlers are programs that traverse the Web 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 and shopping bots crawl the Web to compare prices and products sold by different e-commerce sites.

In this Section we describe how crawlers can be detected by simple pattern matching preprocessor in a form of a log analyzer.

The pre-processing task consists of identifying sessions, extracting features of each session and finally performing session classification.

3.1 Session Identification

The preprocessing task identifies sessions and features that characterize each session. Once a session is defined, the classification algorithm can label it as either belonging to a human user, a well-behaved web crawler or a malicious web crawler.

Session identification is the task of dividing an access log into sessions. Typically, session identification is performed by first grouping all HTTP requests that originate from the same IP address and 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. A drawback of this method is that it is hard to determine a proper threshold-value, as different user-agents exhibit different navigation behaviours. Usually, a 30-min period is adopted as the threshold in Web-mining studies [2]. Due to time constraints of our study, we were only able to employ this simple 30-min threshold to identify sessions. However, regardless of its simplicity, this session identification method has generated fairly successful web crawler classification results in the past (see [4]).\

3.2 Features

The Java-based log analyzer was utilized to pre-process the web server access log file. The log analyzer scans the entries in the log and identifies sessions. 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 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 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 the previous web crawler classification studies, namely [3], [4] and [5], we have identified a list of features that provide distinguishable characteristics between web robots and humans. In our study, the log analyzed extracts the following list of features for each session (Note that in the rest of the paper we will refer to these features based on their numeric ID shown here):

1. Click rate – a **numerical** attributed calculated as the number of HTTP requests sent by a user in a single session. The click rate metric can be used to detect the presence of the web crawlers because higher click rate can only be achieved by an automated script (such as a web robot) and is usually very low for a human visitor of the web site.
2. HTML-to-Image Ratio – a **numerical** attribute calculated as the number of HTML page request 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.
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. Most web crawlers, in order to reduce the amount of data requested from a site, employ the HEAD method when requesting a web page. A human user browsing web site would exclusively request web pages using a GET method instead.
6. Percentage of requests with unassigned referrers – a **numerical** attributed calculated as the percentage of blank or unassigned referrer fields set by a user in a single session. Typically, web crawlers would initiate HTTP requests with unassigned referrer field.
7. 'Robot.txt' file request – a **nominal** attribute with values of either 1 or 0, indicating whether 'robot.txt' file was requested or not requested by a user during a session, respectively. Web administrators, through the Robots Exclusion Protocol, use a special-format file called *robots.txt* to indicate to visiting robots which parts of their sites should not be visited by the robot. For example, when a robot visits a Web-site, say http://www.cse.yorku.ca, it should first check for http://www.cse.yorku.ca/robots.txt. It is unlikely, that any human would check for this file, since there is no link from the Web-site to this file, nor are (most) users aware of its existence.

Generally, as mentioned earlier, in the past research features 1-7 have been good indicators that can help in distinguishing whether the session belongs to a human or a robot. However, based on the recommendations outlined in [9], we have decided to introduce additional features in web robot classification:

8. Standard deviation of requested page's depth – a **numerical** attributed 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 repeated HTTP requests – a **numerical** attribute calculated as the number of repeated requests sent in sequence belonging to the same web directory sent by a user during a session. For instance, a series of requests for web pages matching pattern '/cshome/course/*.* will be marked as consecutive repeated 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 repeated requests.

In [9], authors argue that analytical robot detection techniques must be based on fundamental distinctions between 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 users and well-behaved as well as malicious web robots in server access log sessions.

The importance of features 8 and 9 can be explained as follows. The navigational patterns of humans represent the action of following a series of links on web pages in order to find information, restricted by the link structure of a site. Human patterns may also include frequent back-and-forth navigation through a site, using a Web browser's history, "back", and "forward" feature. Loops may also be present if a human becomes

disoriented during their visit. 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. After an initial crawl of a site, robots are capable of learning precisely where the information that they are seeking resides, so that on repeated visits they may only send requests for specific files or restrict their crawling to specific areas of the site. For the above reasons, the standard deviation of requested pages' depths, i.e. attribute 8, should be low for web robot sessions since a web robot should scan over a narrower directory structure of a web site than a human user. Note that this feature will be effective 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, streaming 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 repeated HTTP requests should be higher in human user sessions and low in web robot sessions.

Therefore, these two new features should be significantly different between various users of a web site. As such, their application in the classification of visitor's sessions should improve the classification accuracy of the results. Note also that in addition to the 9 features listed above, each session stores an additional attribute 'IP Address' that serves as an ID of the session and which is not used in the classification.

Also, 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 in total, is too short to enable labelling. Even by manual inspection, a session with such a few numbers of requests is almost impossible to classify. We are therefore ignoring sessions that are too small (i.e. with less than 5 requests)

3.3 Classification

After the log analyzer parses the log file and extracts the individual sessions, each session and accompanying features are placed in an ARFF file as an instance of a training example. The classifiers employ the generated training datasets to learn the models for classification of web site visitor's sessions.

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

1) Experiment #1: Classification of human sessions and web crawler sessions. For this experiment, we employ the log analyzer to generate a dataset in which a human session is class labeled with value 0 and a session of a known well-behaved web crawler is labeled with value 1. Note that we remove sessions from this dataset that are classified as belonging to malicious web crawlers.
2) Experiment #2: Classification of malicious web crawler sessions. In this experiment, we employ the log analyzer to generate a dataset in which a human session or a session of a known web crawler is class labeled with value 0 and a session of a known malicious web crawler is labeled with value 1.

3.3.1 Classification of well-behaved web crawlers

The classifications described in the first experiment were generated in the following manner. One of the web server access log file is used to form the training dataset. The sessions of the training data set are classified as belonging to a web crawler based on the IP address and user agent field of the user. The log analyzer maintains a table of IP addresses and User Agent fields of all known (malicious or well-behaved) web crawlers (This table can be obtained by visiting website in [10]). If the IP addresses or the user agent field of a training example match the entry in the table, the session is labelled as belonging to a known well-behaved web crawler (class label is set to 0). Otherwise, the session is labelled as belonging to a human visitor (class label is set to 1).

3.3.2 Classification of malicious web crawlers

In the second experiment, the same set of training sessions from the first experiment was used. However, the classification was based on whether the session visitor is a human user / well-behaved web robot or a malicious web robot.

To label a session as belonging to a malicious crawler, we perform the following three tests:

1. First, we make sure that the session does not belong to a well-behaved web robot using the labelling technique described in the first experiment (Namely, by examining the web site in [10]). If there is a match, the test fails and we label the session with value 0.
2. If the labelling is still inconclusive, we attempt to match the IP address or User Agent field of a session with the IP addresses or User Agent field of a known malicious web crawler by consulting list downloaded from the site in [10] (Note that this web site lists both known well-behaved and known malicious web crawlers). If there is a match, we label the session with value 1.
3. If there is still no conclusive label for the session, we perform the third test where we check whether the user identified in the session requests the "robots.txt" file. If this is the case, we label the session with value 1, and 0 otherwise.

In our opinion, these three tests should conclusively decide whether a session should be labelled as belonging to a human visitor, a well-behaved web crawler or a malicious crawler. Firstly, a human visitor cannot pass any of the three tests (Namely, it fails all three tests and the session is correctly labelled with value 0). Secondly, a well-behaved robot would be identified by the first test and such a session would be correctly labelled with value 0. And thirdly, a session belonging to a malicious crawler will be correctly identified either in the second or the third test.

Also, note that we base our classifications on the list of known (malicious or well-behaved) web crawlers listed on the website in [10]. The unknown crawlers, i.e. those that are not listed on the website in [10], are by default labelled as human users. Due to the large number of possible user agent strings, it was impossible to perform pattern matching on all of them. However, the results are not affected since the percentage of unknown crawlers in our training dataset is close to zero.

4. EXPERIMENTAL DESIGN

In the previous Section, we have described how sessions can be classified by a simple log analyzer. In our experimental analysis,

assuming the correctness of classification results derived by the log analyzer, we evaluate the classification accuracy of WEKA classification algorithms. In this section, we outline various details regarding our experimental design.

4.1 Experimental Motivation

The principal reason for conducting the two experiments is to evaluate the classification accuracy of traditional WEKA classification algorithms on datasets containing sessions belonging to human users, known well-behaved web robots and malicious web crawlers.

Additionally, in both experiments we will perform tests with only features 1-7 and with all 9 features and compare the results between the two. Namely, we will examine whether features 8 and 9 can improve the accuracy rate of the classification algorithms.

4.2 Web Server Access Logs

The WEKA data sets were constructed by preprocessing web server access log files provided by York CSE department. The log file belongs to the www.cse.yorku.ca web site domain. In total about 3 million log entries were examined by the log analyzer in this study. The log file used to derive the training dataset contains about four weeks worth of web site activity. Note that in both experiments we utilize the same training dataset. Tables 1 and 2 list the number of sessions and class label counts generated by the log analyzer for experiments 1 and 2, respectively.

- **Table 1: Summary of training dataset used in Experiment #1**

	Training Data Set
Number of Sessions	**47824**
# of Session with Class Label = 0	**46790**
# of Session with Class Label = 1	**1034**

- **Table 2: Summary of training dataset used in Experiment #2**

	Training Data Set
Number of Sessions	**48311**
# of Session with Class Label = 0	**47823**
# of Session with Class Label = 1	**488**

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

122.248.163.1 - - [09/Feb/2010:04:37:38 -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)

As can be observed, the file 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/Feb/2010:04:37:38 -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)). These entries can be employed by the log analyzer to identify visitor sessions.

4.3 Classification Algorithms

The detection of web crawlers was evaluated with the following six classifiers: C4.5, JRIP, Naïve Bayesian, Bayesian Network, k-Nearest Neighbor and LibSVM. The implementation of each algorithm is provided in the WEKA software package. Each classifier is trained on the training dataset with 10-fold cross-validation. In order to determine the classification accuracy, the classification results generated by the classifiers are compared against the 'correct' classifications derived by the log analyzer.

We decided not to evaluate the algorithms based on the misclassification cost. The cost is the same regardless of whether a classification algorithm mislabels a session belonging to a human visitor, a well-behaved web crawler or a malicious crawler. Namely, the security defense systems used to protect access to the Internet web sites must allow legitimate and deny illegitimate sources access to the data. This fact implies that the same cost can be assigned to both false positive and false negative cases. Therefore, the misclassification cost analysis is not applicable in our study.

Also, due to high class imbalance, the up-sampling pre-processing techniques were used to improve the classification accuracy. Various up-sampling ratios were tried for each algorithm and one that produced the best classification accuracy was chosen as the representative classification result for that algorithm.

4.4 Experimental Parameters

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. If you examine the ratio of class labels in Tables 1 and 2, you can see that we have this exact problem with our datasets, i.e. class imbalance problem.

4.4.1 Recall, Precision and F_1 score

In order to test the effectiveness of our classifiers, we adopted metrics that are commonly applied to imbalanced datasets: recall, precision, and the F_1-score [4], which summarizes both recall and precision by taking their harmonic mean. F_1 score summarizes the two metrics into a single value, in a way that both metrics are given equal importance. 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:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+False Negative}} * 100 \quad (1)$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}} * 100 \quad (2)$$

$$F_1 = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (3)$$

Positive classification, in our study is the classification of a session as crawler (the target class). The above formulae therefore translate to:

$$\text{Recall} = \frac{\text{\# of crawler sessions correctly classified}}{\text{\# of actual crawler sessions}}$$

$$\text{Precision} = \frac{\#\text{ of crawler sessions correctly classified}}{\#\text{ of predicted crawler sessions}}$$

4.4.2 Information Gain, Gain Ratio and Significance of the Difference Test

Additionally, we rank the most important features by employing attribute selection methods such as information gain and gain ratio. The ranking provides the purity test of the two proposed features.

The effectiveness of the new attributes in classifying visitor's sessions can be evaluated further by applying the significance of difference test or t-test. Namely, we separate the sessions in 2 groups, true negatives and false positives. The sessions are grouped into true negatives if both the log analyzer and classification algorithms label the session with class label 0. The second set of sessions are grouped into false positives if the classification algorithms labels these sessions with class label of 1 and if the log analyzer labels the same session with the opposite class label of 0 ..

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 95% confidence interval. The calculation of the significance of the difference test is the following:

$$t = \frac{|mean_1(f) - mean_2(f)|}{\sqrt{\frac{Var_1(f)}{n_1} + \frac{Var_2(f)}{n_2}}} \quad (4)$$

In the equation above $mean_1$ 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 [11].

By applying the significance test, we can test whether features 8 and 9 are significantly different between the two groups of sessions. If the difference is significant, this fact provides additional proof that two features are valuable in classifying user sessions.

5. CLASSIFICATION RESULTS

In this Section we analyze the results of our two experiments. In Sections 5.1 and 5.2, we present and discuss the results derived in Experiments 1 and 2, respectively. In Section 5.3, some additional observation and discussion of results is given.

5.1 Experiment 1

In this section, we present the results derived in the first experiment. The motivation for this experiment was to evaluate whether features 8 and 9, i.e. consecutive repeated request rate and standard deviation of requested page depths, can improve the accuracy in classifying sessions as either belonging to a human user or a well-behaved web crawler.

5.1.1 Classification Accuracy

The Figure 2 shows the accuracy rate when the seven classification algorithms are trained on the data set containing only features 1-7. Figure 3 displays the accuracy rate when the seven classification algorithms are trained on the data set 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 in both Figures 2 and 3. Figure 4

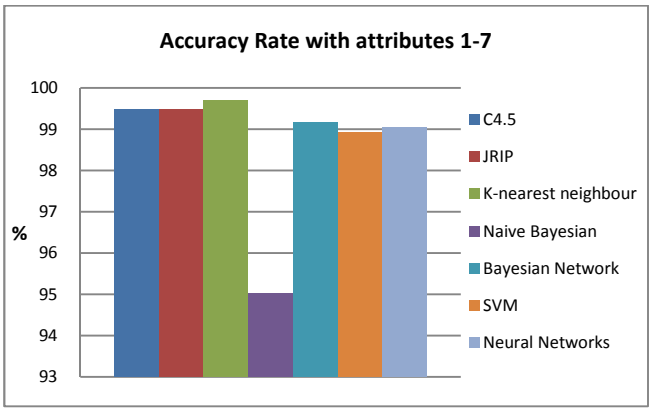

Figure 2: Classification accuracy rate for various classifiers trained on the dataset that contains only features 1-7

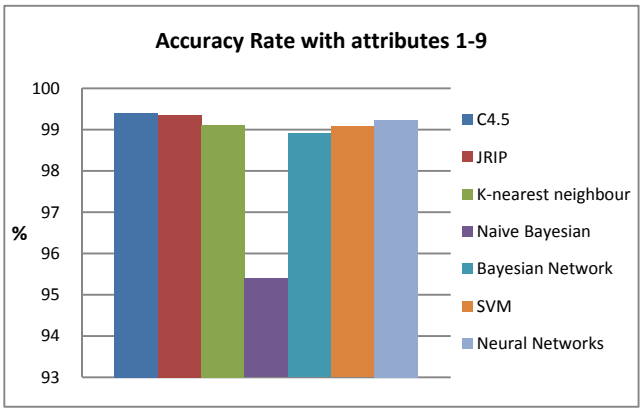

Figure 3: Classification accuracy rate for various classifiers trained on the dataset that contains all 9 features

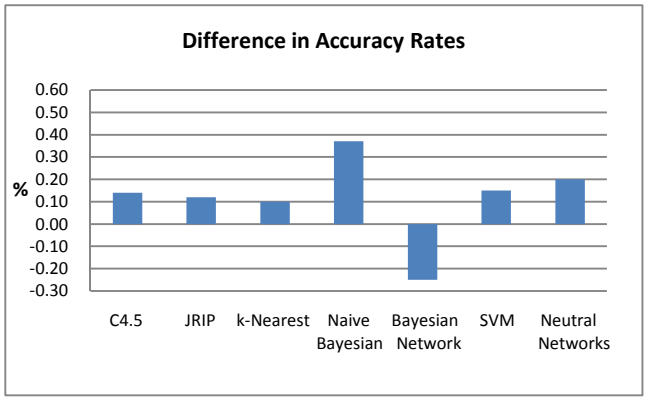

Figure 4: Difference between accuracy rates in Figures 2 and 3 relative to the results in Figure 2

shows the difference in terms of percentage points between the accuracy rates of Figure 2 and Figure 3 relative to results shown in Figure 2. 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.

5.1.2 Recall, Precision and F_1 score

A more accurate evaluation of classifiers can be derived by examining the recall, precision and F_1 score metrics. Figure 5 displays the recall, precision and F_1 score for the seven algorithms

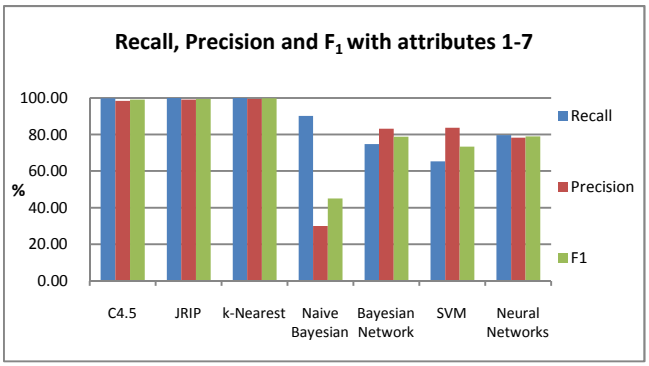

Figure 5: Recall, Precision and F1 score for various classifiers trained on the dataset that contains only features 1-7

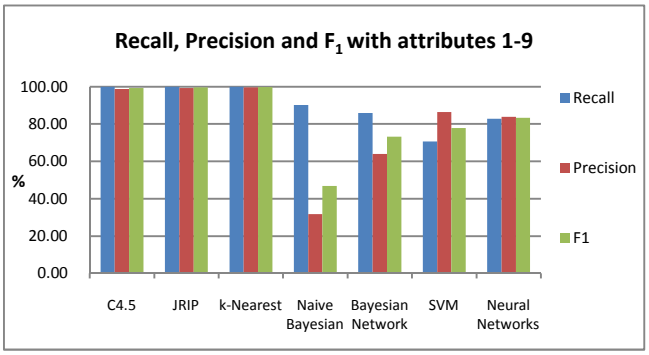

Figure 6: Recall, Precision and F_1 score for various classifiers trained on the dataset that contains all 9 features

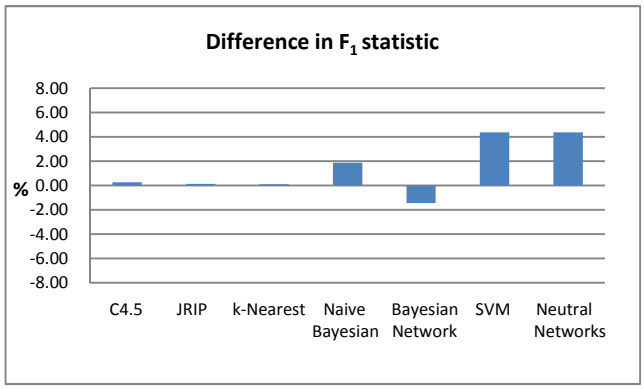

Figure 7: Difference between F_1 score in Figures 5 and 6 relative to the F_1 score in Figure 5 for all seven algorithms

that are trained on the data set containing only features 1-7. Figure 6 displays the recall, precision and F_1 score for the seven algorithms that are trained on the data set containing all 9 features. Finally, Figure 7 shows the difference in terms of percentage points between the F_1 score of Figures 5 and 6 relative to results shown in Figure 5. As can be observed, in almost all seven algorithms (except Bayesian Network), the F_1 score is higher. This means that harmonic average between the recall and precision is higher if all 9 features are used to train the classification algorithms.

5.1.3 Entropy-based Attribute Ranking

Lastly, Table 3 shows the ranking between all 9 features in terms of information gain and gain ratio metrics. As expected, the attribute that marks whether robots.txt file was requested during a session is at the top of the ranking for both metrics. This observation is expected since web crawlers that are legitimate and well-behaved should access the robots.txt file every time they visit a site. The percentage of unassigned referrers is another attribute that defines whether a session belongs to a web crawler. The referrer parameter should be only assigned by a browser of the user visiting the web site and should be left blank if the visitor is a web crawler. Therefore, this attribute should have high percentage values if the session belongs to a web crawler and low percentage values if the user is a human user accessing the site via a browser.

Table 3: Attribute ranking in terms of Information Gain and Gain Ratio metrics (ordered top down from best to worst)

Information Gain	Gain Ratio
'robots.txt' is requested	'robots.txt' is requested
% of Unassigned Referrers	% of Unassigned Referrers
% of Repeated requests	% of Repeated requests
Std. Dev. of Page Depth	% of PDF documents
% of Error requests	% of HEAD requests
% of PDF documents	Std. Dev. of Page Depth
Click Rate	% of Error requests
HTML to Image ratio	Click Rate
% of HEAD requests	HTML to Image ratio

The two features that we have introduced in this study are near the top of the rankings. The percentage of consecutive repeated 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 repeated requests (in the same directory of the web site) can only be attributed to a human user.

The standard deviation of the page depth of requests is also fairly important although less so than the consecutive repeated requests feature. As explained in Section 3.1.2, web crawlers should have lower standard deviation of page depth than the requests made by human users.

The significance of the differences of values between two groups of sessions for features 8 and 9 can be confirmed by applying the significant difference of the mean test (Equation 4 or the t-test) [11]. The test compares the mean values of features 8 and 9 for sessions classified as well behaved web crawler's sessions and sessions classified as human's sessions. The significance difference test proves that in the training dataset, the difference between two groups of sessions in terms of the mean value of features 8 and 9 is significantly different (t value in the Equation 4 is above 1.96 with degrees of freedom set to 47823). Therefore, the two novel features can be very helpful in classifying well-behaved web robot sessions.

5.2 Experiment 2

In this section, to we present the results relevant to the second experiment conducted in our study. The motivation for this experiment was to evaluate whether features 8 and 9, i.e. consecutive repeated request rate and standard deviation of page depth, can improve the accuracy in classifying sessions as belonging to malicious web crawlers.

5.2.1 Classification Accuracy

The Figure 8 shows the accuracy rate when the seven classification algorithms are trained on the data set containing

only features 1-7. Figure 9 displays the accuracy rate when the seven classification algorithms are trained on the data set containing all 9 features. As expected, due to class imbalance the classification accuracy is very high (at 96% or above) for all seven classification algorithms in both Figures 8 and 9. The classification accuracy is higher than in Experiment 1 since in the Experiment 2 there is even greater class imbalance. Figure 10 shows the difference in terms of percentage points between the accuracy rates of Figure 8 and Figure 9 relative to results shown in Figure 8. As can be observed, there is a slight improvement in accuracy rate when all 9 features are used for all algorithms except in the scenario where SVM algorithm is employed which shows a slight decline in accuracy.

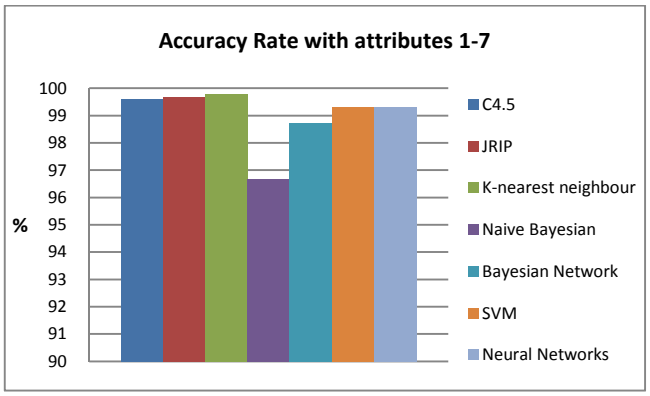

Figure 8: Classification Accuracy Rate for various classifiers trained on the dataset that contains only attributes 1-7

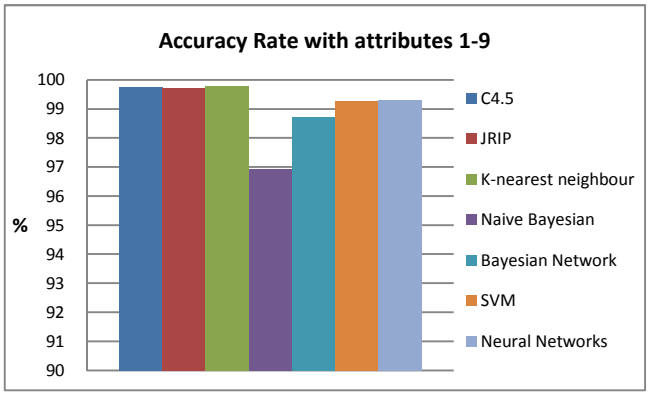

Figure 9: Classification Accuracy Rate for various classifiers trained on the dataset that contains all 9 attributes

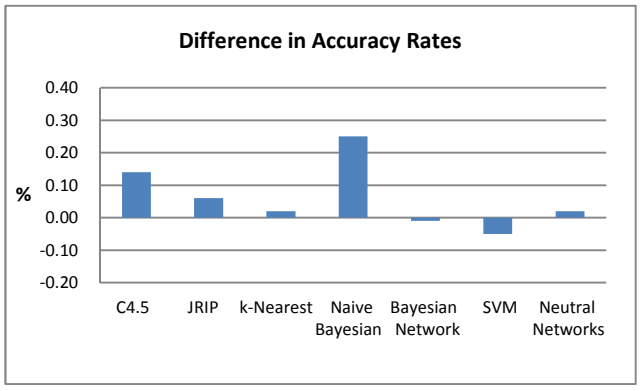

Figure 10: Difference between accuracy rates in Figures 8 and 9 relative to the results in Figure 8

5.2.2 Recall, Precision and F_1 score

A more accurate evaluation of classifiers can be derived by examining the recall, precision and F_1 score metrics. Figure 11 displays the recall, precision and F_1 score for the seven algorithms that are trained on the data set containing only features 1-7. Figure 12 displays the recall, precision and F_1 score for the seven algorithms that are trained on the data set containing all 9 features. Finally, Figure 13 shows the difference in terms of percentage points between the F_1 score of Figures 11 and 12 relative to results shown in Figure 11. As can be observed, in almost all seven algorithms (except for Bayesian Network and SVM) the F_1 score is higher. This means that harmonic average between the recall and precision is higher if all 9 features are used to train the classification algorithms.

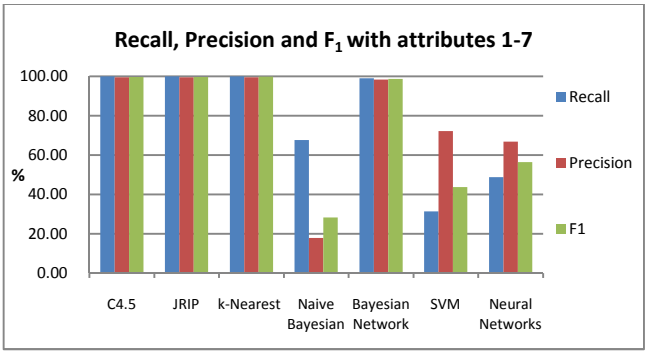

Figure 11: Recall, Precision and F1 score for various classifiers trained on the dataset that contains only attributes 1-7

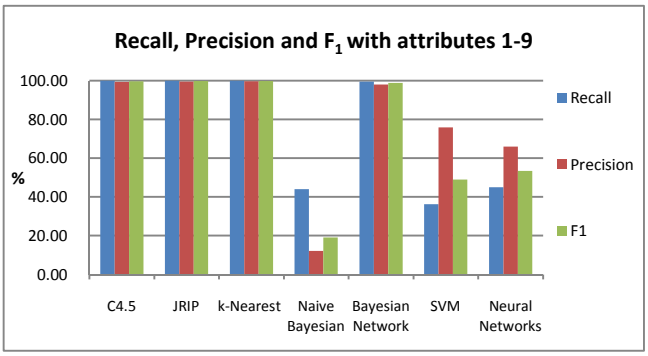

Figure 12: Recall, Precision and F_1 score for various classifiers trained on the dataset that contains all 9 features

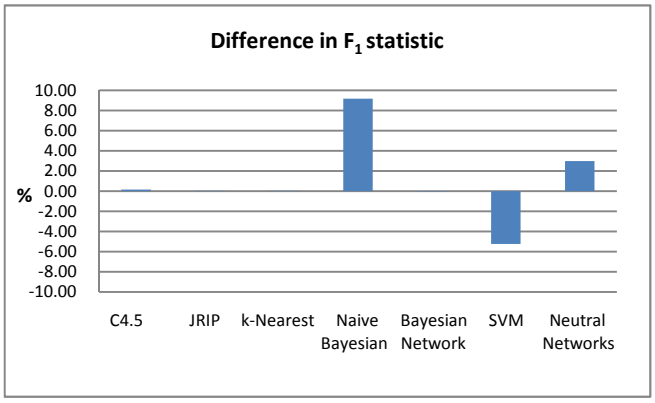

Figure 13: Difference between F_1 score in Figures 11 and 12 relative to the F_1 score in Figure 11 for all seven algorithms

5.2.3 Entropy-based Attribute Ranking

Lastly, Table 4 shows the ranking between all 9 features in terms of information gain and gain ratio metrics. Similar feature rankings are observed in Experiment #2 in comparison to rankings from Experiment #1 shown in Table 3. Again, two features, the request for robots.txt file and the rate of unassigned referrers in a session, are ranked 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 as well. The consecutive repeated request feature is again fairly high in the rankings. Also, the results show that the standard deviation of the page depth of requests is quite important in defining the session as belonging to a malicious web crawler.

This hypothesis can be further confirmed by applying the significant difference of the mean test (Equation 4 or the t-test) [11]. The test compares the mean values of features 8 and 9 for sessions classified as malicious web crawler's sessions and sessions classified as human's of well-behaved web crawler's sessions. The significance difference test proves that in the training dataset, the difference between two groups of sessions in terms of the mean value of features 8 and 9 is significantly different (t value in the Equation 4 is above 1.96 with degrees of freedom set to 48310). Therefore, the two novel features can be very helpful in classifying malicious web robot sessions.

Table 4: Attribute ranking in terms of Information Gain and Gain Ratio metrics (ordered top down from best to worst)

Information Gain	Gain Ratio
'robots.txt' is requested	'robots.txt' is requested
% of Unassigned Referrers	% of Error requests
Std. Dev. of Page Depth	% of Unassigned Referrers
Html to image ratio	Std. Dev. of Page Depth
% of Repeated requests	% of Repeated requests
Click Rate	Click Rate
% of Error requests	Html to image ratio
% of PDF documents	% of PDF documents
% of HEAD requests	% of HEAD requests

5.3 Discussion and Additional Observations

The results presented in the previous section show that most of the classifiers achieve very high recall and precision scores in both experiments. The classification results derived with C4.5, JRIP and k-Nearest Neighbor algorithms have the recall, precision and F_1 metric scores well over 90%, in both experiments.

The classification results of experiment 1 are expected. As we have discussed previously, the characteristics of web site usage by well-behaved web crawlers should significantly differentiate from the usage by human users in terms of the features examined in this study. Namely, the classification algorithms can detect the difference between the features used to describe the well-behaved web crawlers and human users.

However, the classification results of experiment 2 are interesting. Namely, the same three algorithms in question, C4.5, JRIP and K-nearest neighbor can separate malicious robots from well-behaved robots and human users with very high precision and recall. This result implies that the values of the features used to describe the malicious crawlers are significantly different from the values of the features used to describe the well-behaved crawlers.

5.3.1 Unknown crawlers

The classification results are promising. Overall, the malicious and well-behaved robots can be detected by standard data mining classifiers. However, the detection of unknown malicious robots in addition to known malicious crawlers would be even more helpful in mitigating the denial of service problem. Therefore, we performed manual examination of the classification results derived in experiment 2 to see if the classifiers can detect unknown (and potentially malicious) crawlers.

Firstly, we label a session as belonging to an unknown crawler if the user agent string associated with that session is not included in the table of user agent fields on the website in [10] and also if the user agent string contains the patterns `bot` or `crawl.*` as substrings. We assume that the log analyzer would mislabel such a session. In fact, as mentioned in Section 3, the log analyzer would predict that such a session belongs to a human user or well-behaved web crawler.

Therefore, assuming that classification algorithms can correctly classify a session as belonging to an unknown web crawler, we examined the false positive cases. A single training example is marked as a false positive if the classification algorithm labels the session as belonging to a malicious web crawler while the log analyzer labels the same session as belonging to a human user or well-behaved web crawler. By manually examining these false positive cases for C4.5 and JRIP algorithms (the two algorithms with the highest accuracy rates in terms of F_1 score), we discovered that some[1] sessions that were mislabeled by the log analyzer as belonging to a human user or well-behaved robot were in fact unknown robots. This is an interesting observation since it provides some indication that sessions of known malicious crawlers could be used to detect the presence of unknown and possibly malicious crawlers by employing data mining classifiers.

6. CONCLUSIONS 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 both known well-behaved web crawlers and malicious web crawlers using existing data mining classification algorithms.

Firstly, we derive the training dataset by employing the log analyzer as a session classifier. Each training example consists of seven standard features employed in previous studies on web crawler classification. However, we also introduce two new features for improving classification of user sessions.

Then in the second part of our study, we perform two experiments. In the first experiment, by applying seven different data mining classifiers, we classify user sessions as belonging to human visitors and known web crawlers. In the second experiment we perform further classification and label sessions as either belonging to human visitors/known web crawlers or malicious web crawlers. In both experiments, the classifications generated by the log analyzer in the first part of our study are compared against the classification results generated by the seven different classifiers. Due to high class imbalance in the datasets, we adopted metrics that are commonly applied to imbalanced datasets such as recall, precision and F_1 score to evaluate the accuracy of classification algorithm results. In addition to

[1] However, some sessions were actually human users which classifiers just incorrectly classified.

classification accuracy evaluation, we further rank all the features in the datasets using information gain and gain ratio metrics.

The following three general conclusions were derived from the experimental evaluation:

- The classification accuracy of classification algorithms such as C4.5, JRIP and k-Nearest Neighbor is very high. The recall and precision values in both experiments are at least above 90%.
- The two new features proposed, the consecutive repeated 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.
- Additionally, the new features improve the classification results in both experiments in terms of accuracy, recall, precision and F_1 score.
- Also, evidence given in Section 5.3, provide some proof that the new features can detect the presence of unknown web crawlers as well.

Based on the above results, the classification algorithms still experience fairly high rates of false negative and false positives. Some network intrusion detection systems require almost non-existent misclassification. However, the results are promising and we believe with customization of either C4.5 or JRIP, the misclassification rates of known well-behaved and malicious web crawlers can surely be reduced. For instance, a condition such as "if robots.txt file was requested than classify the sessions as belonging to a web crawler" can be hardcoded inside the C4.5 algorithm. Other rules based on unassigned referrer and sequence of repeated requests can be hardcoded as well to improve the classification results.

As evident in our study, the characteristics between the web crawlers (both malicious and well-behaved) and human users are significantly different. As such they can be used by the classification algorithms to derive correct classification models. However, the classification of crawlers that attempt to mimic human users will remain the most difficult future classification challenge.

7. ACKNOWLEDGMENTS

We would like to thank, Professor Aijun An, for approving this interesting research topic. We would also like to thank our network administrator, Ulya Yigit, for providing the web server access log files for our experiments.

8. 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] (2010, Dec.) WEKA. [Online]. http://www.cs.waikato.ac.nz/ml/weka/

[3] P.-N. Tan and V. Kumar, "Patterns, Discovery ofWeb Robot Sessions Based on their Navigation," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 9-35, Jan. 2002.

[4] A. Stassopoulou and M. D. Dikaiakos, " Web robot detection: A probabilistic reasoning approach," *Computer Networks: The In-ternational Journal of Computer and Telecommunications Networking*, vol. 53, no. 3, pp. 265-278, Feb. 2009.

[5] C. Bomhardt, W. Gaul, and L. Schmidt-Thieme, "Web Robot Detection - Preprocessing Web Logfiles for Robot Detection," in *In Proc. SISCLADAG*, Bologna, Italy, 2005.

[6] L. Wei-Zhou and Y. Shun-Zheng, "Web robot detection based on hiddenMarkovmodel," in *In Proceedings of international conference on communications, circuits and systems*, Guilin, China, 2006, pp. 1806-1810.

[7] L. v. Ahn, M. Blum, J. Langford, and N. Hopper, "CAPTCHA: using hard AI problems for security," in *In: Proceedings of Eurocrypt*, Warsaw, Poland, 2003, pp. 294-311.

[8] X. Lin, L. Quan, and H. Wu, "An automatic scheme to categorize user sessions in modern HTTP traffic," in *In Proceedings of IEEE global telecommunications conference*, New Orleans, Louisiana, 2008, pp. 1-6.

[9] D. Doran and S. S. Gokhale, "Web robot detection techniques: overview and limitations," *Data Mining and Knowledge Discovery*, pp. 1-28, Jun. 2010.

[10] (2010, Dec.) User-Agents.org. [Online]. http://www.user-agents.org/index.shtml?t_z

[11] R. Wonnacott and T. Wonnacott, *Introductory Statistics*, 4th ed. USA: John Wiley and Sons, 1996.