Journal of Data Analysis and Information Processing, 2015, 3, 1-10
Published Online February 2015 in SciRes. http://www.scirp.org/journal/jdaip
http://dx.doi.org/10.4236/jdaip.2015.31001

Agglomerative Approach for Identification and Elimination of Web Robots from Web Server Logs to Extract Knowledge about Actual Visitors

Dilip Singh Sisodia[1], Shrish Verma[2], Om Prakash Vyas[3]

[1]Department of Computer Science & Engineering, National Institute of Technology, Raipur, India
[2]Department of E & TC, National Institute of Technology, Raipur, India
[3]Department of IT, Indian Institute of Information Technology, Allahabad, India
Email: dssisodia.cs@nitrr.ac.in

Received 20 January 2015; accepted 10 February 2015; published 19 February 2015

Abstract

In this paper we investigate the effectiveness of ensemble-based learners for web robot session identification from web server logs. We also perform multi fold robot session labeling to improve the performance of learner. We conduct a comparative study for various ensemble methods (Bagging, Boosting, and Voting) with simple classifiers in perspective of classification. We also evaluate the effectiveness of these classifiers (both ensemble and simple) on five different data sets of varying session length. Presently the results of web server log analyzers are not very much reliable because the input log files are highly inflated by sessions of automated web traverse software's, known as web robots. Presence of web robots access traffic entries in web server log repositories imposes a great challenge to extract any actionable and usable knowledge about browsing behavior of actual visitors. So web robots sessions need accurate and fast detection from web server log repositories to extract knowledge about genuine visitors and to produce correct results of log analyzers.

Keywords

Web Robots, Web Server Log Repositories, Ensemble Learning, Bagging, Boosting, and Voting, Actionable Knowledge, Usable Knowledge, Browsing Behavior, Genuine Visitors

1. Introduction

Web robots are autonomous agents used to browse the web in a mechanized and organized manner. They are

How to cite this paper: Sisodia, D.S., Verma, S. and Vyas, O.P. (2015) Agglomerative Approach for Identification and Elimination of Web Robots from Web Server Logs to Extract Knowledge about Actual Visitors. *Journal of Data Analysis and Information Processing*, **3**, 1-10. http://dx.doi.org/10.4236/jdaip.2015.31001

also known as web crawlers, spiders, wanderers, and harvesters etc. They start their working with seed URLs lists and recursively visit hyperlinks accessible from that list. The purpose of crawlers is to discover and retrieve content and knowledge from the web on behalf of various web-based systems and services. Web robots are mostly used by search engines as a resource sighting and reclamation tools, which performs a vital role of maintaining data repositories state-of-the-art by powering the swift discovery of resources on the Internet. To sustain with the huge volumes of time-sensitive information, they must perform comprehensive searches, tender focused functionality and frequent visits to servers. These prospects have led to a spectacular augmentation in the number and types of robots, and their ferocity in visiting servers [1].

Robots can be classified on the basis of their prime functionality. For example Indexers (or 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. Analyzers (or shopping bots) crawl the web to compare prices and products sold by different e-Commerce sites. Experimental (focused crawlers) seek and acquire web-pages belonging to pre-specified thematic areas. Harvesters (email harvesters) collect email addresses on behalf of email marketing companies or spammers. Verifier (site-specific crawlers) performs various website maintenance chores, such as mirroring web sites or discovering their broken links. RSS crawler use to retrieve information from RSS feeds of a web site or a blog. Scrapers used to automatically create copies of web sites for malicious purposes [2]. Since their inception (First web robots were introduced in 1993) they are increasing exponentially. Because they are very simple to create and they offer great job by circumvent collection of information.

While many of the robots are legitimate and do serve a significant role to find relevant content on the web, but there are robots which do not proceed to a noteworthy reason. There are some instances where role of web robots are not defensible because it may harm to others by any extent and the several reasons are as follows: first the amount of traffic caused by crawlers may drain both the computation and communication resources of a server. Second many e-commerce web sites may not wish to serve incoming HTTP requests from unauthorized web crawlers to maintain business secrecy with competitor. Third commercial web portals perform analysis of browsing behavior of their customers and visitors and understand about their geographic, demographic trends but results are highly inflated due to presence of web robots traffic. Fourth pay-per-click advertising can be seriously harmed by deceptive behavior of web robots which involves among other things the unwilling or malicious repetitive “clicking” on advertisement links by Web robots [1] [3]. Hence it is extremely enviable to recognize visits by web robots and discriminate it from human users.

This paper is organized as follows: in Section 2 the previous work on web robot detection is discussed. In Section 3, outline of the brief description of methodology adopted for this work is presented. In Section 4, we describe the experimental design and. In Section 5, experiments are performed and experimental results were discussed. In Section 6 the paper has concluded with proposed future work.

2. Related Work

A substantial amount of literature has been published on identification of web robots traffic from web server access logs. In [4] author presents a comprehensive survey of web robots detection techniques, these techniques are broadly classified in to two category offline and real time. Real time web robot detection techniques do not operate over web server access logs and beyond the scope of this discussion. Offline web robot detection techniques works on web server logs and further classified in to syntactic log analysis, traffic pattern analysis and analytical learning. Syntactic log analysis is very simple and mostly relies on parsing of unequivocally documented information in web server logs but it can only detect already known robots. In traffic pattern analysis underlying assumption is that web robots navigational behavior is inherently different from human users and web robots are detected on this expected line rather than the knowledge of their name and origin.

Analytical learning techniques are considered more robust to handle camouflage and evasive behavior of robots as compare to others because it rely on learning features of robot traffic on the server. In [1] author use navigational behavioral patterns of web robots and humans to derive different features from each session and a decision tree (C4.5) to classify sessions based on a feature vector of their characteristics. In their study, they were able classify web robots with 90% accuracy after examining at least 4 requests within any given session. In [5] Stassopoulou *et al.* use a probabilistic learning based Bayesian network to identify Web robots by using number of features to achieve an 85% to 91.4% classification accuracy amongst a variety of datasets. In [6] Bomhardt *et al.* use a neural network model and a feature vector of session characteristics to detect web robots and compare

results with [1] by using the many overlapping features. In [7] Lu *et al.* consider a hidden Markov model to distinguish robot and human sessions based on request arrival patterns and claims detection rate of 97.6%. Apart from the above analytical learning methods of robot detection other example works are as follows. In [8] Guo *et al.* proposed two detection algorithms that consider the volume and rate of resource requests to a Web server. In [9] Park *et al.* use a human activity detection approach of identifying web robots and web browsers by getting the evidence of mouse movement or keyboard typing from the client. All above mentioned major contributions in this field are suffer from poor session labeling and benefits of ensemble learning or agglomerative techniques. In the view of above, in this work we well addressed these two short comings of these methods.

3. Methodology

In **Figure 1** we outline the proposed methodology to carry out the experiments. In following sections we briefly describe important work to be performed in different steps.

3.1. Web Server Access Logs

A Web server access log store the detailed information of each request made from user's web browsers to the web server in a chronological order [10]. An example of classic web server log entries is given as follows and brief description in **Table 1**.

```
11.111.11.111 - - [15/Dec/2013:00:01:02 -0800] "GET
/forum/member.php?45067-Carla-Zenis&tab=activitystream&type=all HTTP/1.1" 200 10463
"http://www.google.com/bot.html" "Mozilla/5.0 (compatible; Googlebot/2.1)"
```

3.2. Session Identification

Any user can visit the particular website many times during a specific time period. Session identification aims at dividing the multi visiting user sessions into single ones. But, due to inherent constraints of the HTTP protocol (*i.e.* HTTP is stateless and connectionless protocol), these records are incomplete. We cannot assign distinctively all the requests contained by web server logs to the entity that has performed them. So discovering the user sessions from web server logs is a multifaceted and tricky task. Our log analyzer uses session-duration heuristic

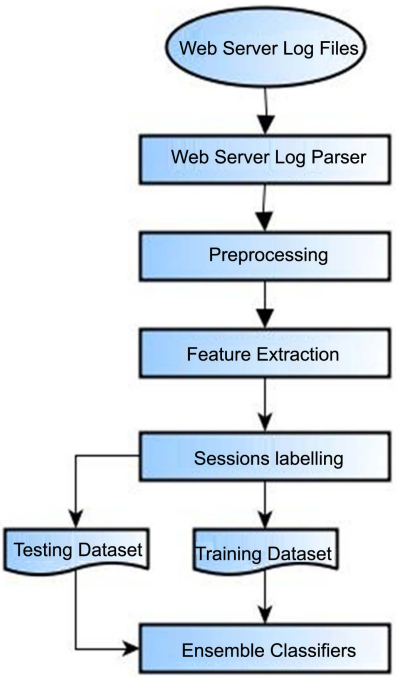

Figure 1. Proposed methodology.

Table 1. Brief description of web log entry headers.

Log entry headers	Description of headers
11.111.11.111	Remote host address
-	Remote log name
-	User name
15/Dec/2013:00:01:02	Timestamp
-800	Time zone of the request
GET	Request method
/forum/member.php?45067-Carla-Zenis&tab=activitystream&type=all	Path on the server
HTTP/1.1	Protocol version
200	Service status code
10463	Size of the returned data
http://www.google.com/bot.html	*Referrer
Mozilla/5.0 (compatible; Googlebot/2.1)	*User agent

*These fields only appear in the extended log file (ELF).

in conjunction with same user agent in which total session duration may not exceed a threshold θ and described by the same user agent string. Given t_0, the timestamp for the first request in a constructed session S, the request with a timestamp t is assigned to S, if $t - t_0 \leq \theta$. This heuristics varies from 25.5 minutes to 24 hours while 30 minutes is the mostly used default timeout for session duration [11] [12].

3.3. Feature Extraction

In this step our log analyzer will extract the values of different features from identified sessions. These features are based on assumption that browsing behavior of web robots is different from human users and very useful in distinguishing between human users and web robots. All these features are adopted from [1] and describe **Table 2**.

3.4. Session Labeling

After extraction and selection of highly relevant features our log analyzer labels the all web serve log sessions in to two classes' *i.e.* Human visitor sessions or web robot sessions. For robust session labeling we adopt a multi-fold approach in which first step we applied well known heuristics based session labeling algorithm of [1] but this will lead to poor session labeling. So in second step our log analyzer uses the database of IP addresses and user agent fields of well known bots [13]. If the web serve log session's IP addresses or user agent is matches with IP or user agent of well known crawlers then session is labeled as web robot sessions.

3.5. Ensemble Based Methods

Ensemble is a combination of multiple classifiers so as to improve the generalization ability and increase the prediction accuracy [14]. The most popular combining techniques are boosting, bagging and voting. In bagging, each model in the ensemble votes with equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly drawn subset of the training set [15]. Whereas, in boosting, each classifier is dependent on the previous one, and focuses on the previous one's errors. Examples that are misclassified in previous classifiers are chosen more often or weighted more heavily [16]-[18]. Voting is a combining strategy of classifiers. Majority Voting and Weighted Majority Voting are more popular methods of Voting [19] [20].

4. Experimental Design

In the previous section we have explained the methodology followed for these experiments. In running the experiments, we used the open source implementation of machine learning algorithms [21]. We used bagging [15], boosting [17], and voting ensemble [20] algorithms as implemented in [21] with default parameters. Bagging

Table 2. Brief description of extracted features from web server logs.

S. No.	Name of Features extracted from sessions	Type of Features	Remarks
1	Total Hits	Numeric	Total number of HTTP requests sent by a user in a single session
2	% Image	Numeric	Percentage of Images files Requests (*i.e.* .jpg, .png, .tiff, .ico, .gif etc.)
3	% HTML	Numeric	Percentage of HTML files Requests (*i.e.* .html.php, .htm, .js, .cgi etc.)
4	% Binary doc	Numeric	Percentage of Binary Doc files Requests (*i.e.* .doc, .pdf, .ps, .xls, .ppt etc.)
5	% Binary Exe	Numeric	Percentage of Binary Exe files Requests (*i.e.* .cgi, .exe, .dll, .dat, .jar etc.)
6	% ASCII	Numeric	Percentage of Ascii files Requests (*i.e.* .txt, .cpp, .java, .xml, .c etc.)
7	% Zip	Numeric	Percentage of Compressed files Requests (*i.e.* .zip, .rar, .gzip, .tar, .gz etc.)
8	% Multimedia	Numeric	Percentage of Multimedia files Requests (*i.e.* .mp3, .mp4, .wmv, .avi, .mpg etc.)
9	% Other File	Numeric	Percentage of Other file format Requests (*i.e.* .swf, .com, .css etc.)
10	Bandwidth	Numeric	Number of bytes requested from server
11	Session Time	Numeric	Approximated total time of the session in seconds
12	avgHTML Req Time	Numeric	Average Time between two HTML requests
13	Total Night Time Req	Numeric	Total time for request made between 12 am and 7 am
14	total Reapeated Req	Numeric	Fraction of repeated requests
15	% Errors	Numeric	Percentage of Requests with status ≥ 400
16	% GET	Numeric	Percentage of Requests made with GET method
17	% POST	Numeric	Percentage of Requests made with POST method
18	% Other Method	Numeric	Percentage of Requests made with Other methods
19	Depth Of Traversal	Numeric	Depth of traversal
20	Length Of Session	Numeric	Session length
21	Is Robotstxt Visited	Nominal	Indicates whether robots.txt file visited or not
22	% HEAD	Numeric	Percentage of Requests made with HEAD method
23	% Unassigned Referrer	Numeric	Percentage of Requests made with Unassigned Referrer (or Referrer = "–")

and boosting are implemented with C4.5 classifier (implemented as J48) in [21]. The base classifiers used for voting are C4.5 [22], random forests [16] and Naive Bayes [22] which are popular in web robot detection [1] [5]. The combination rule of vote is majority voting. The data sets described below in Section 4.1 is used to test the performance of various ensemble methods. 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

4.1. Dataset Description

Brief descriptions of data sets are used for this study are present in **Table 3**.

4.2. Performance Evaluation

The effectiveness of the classifier can be evaluated using different performance parameters. If TP and TN are total number of correctly identified true positive samples and true negative samples respectively, and FP and FN are total number of correctly identified false positive samples and false negative samples respectively, In a two-class problem (web robots, humans) the confusion matrix (shown in **Table 4**) records the results of correctly and incorrectly recognized examples of each class. In this study classification of user sessions as web robot

Table 3. Data set description.

Data sets	Remarks	Human Sessions	Robot Sessions	# of Instances	Size(in KB)	# of training instances	# of testing instances
DS1	Session length ≥ 3	31017	2361	33378	2634	30041	3337
DS2	Session length ≥ 4	22026	1836	23862	1982	21476	2386
DS3	Session length ≥ 5	16100	1557	17657	1481	15892	1765
DS4	Session length ≥ 6	12497	1201	13698	1155	12329	1369
DS5	Session length ≥ 7	10126	996	11122	975	10010	1112

Table 4. Confusion or contingency matrix.

Actual Instances	Predicted Instances	
	Web robots sessions	Human sessions
Human sessions	Human sessions incorrectly classified as robots (FP)	Human sessions correctly classified as humans (TN)
Web robots sessions	Web robots sessions correctly classified as robots (TP)	Web robots sessions incorrectly classified as humans (FN)

sessions is considered as positive class and human sessions as negative class. Given a classifier and a session, there are four possible outcomes. If the session is robot session and it is classified as robot session, it is counted as a true positive (TP); if it is classified as human session, it is counted as a false negative (FN). If the session is human session and it is classified as human session, it is counted as a true negative (TN); if it is classified as robot session, it is counted as a false positive (FP) [23]. So the contingency matrix will convert as follows.

This matrix forms the basis for many classifier evaluation measures but it is evident from **Table 3** that robot data set is highly skewed towards one class. Due to this class imbalance problem the performance of the classifier simply measured by accuracy can be misleading. Therefore to test the effectiveness of ensemble classifiers we adopted recall, precision and f1-measures, which is harmonic mean of recall and precision to give equal importance to both and used to panelize a classifier which gives high recall but scarifies precision and vice-versa [24]. The equations are given in (1) to (3).

$$\text{Recall}(\text{R}) = \frac{\#\text{ of web robot sessions correctly classified}}{\#\text{ of actual web robot sessions}} \tag{1}$$

$$\text{Precision}(\text{P}) = \frac{\#\text{ of web robot sessions correctly classified}}{\#\text{ of predicted web robot sessions}} \tag{2}$$

$$\text{F1}-\text{Measure}(\text{F}) = \frac{2\text{RP}}{\text{R}+\text{P}} \tag{3}$$

5. Experimental Results

The objective for this experiment was twofold first to test the effectiveness ensemble leaner's in robot session identification with comparison of simple learners. Second to evaluate whether session length can improve the accuracy of web robot detection. We present and discuss the results of our two experiments in following sections.

5.1. Experiment-I

Here we present the results derived from first experiment. The comparisons of the recall, precision and F1 scores of Ensemble learners (boosting, bagging, and voting) with simple classifiers are shown in **Figure 2** & **Figure 3**,

for human sessions and web robots respectively. It is evident from the presented graphs that Ensemble learners produced improved value of recall, precision and F1 scores, as compared to simple classifiers especially with Naïve Bayes. Despite our main objective being only to web robot session identification but we evaluated precision, recall and F1-measure for both majority and minority class as human sessions and web robot sessions respectively just to see the impact of majority class on different measures. As our data set is dominated by human sessions (majority class) (**Figure 2**) has perfect recall (100%) but lower precision (as expected) and F1 score yields the value closer to the smaller of two. The robot sessions represent the minority class in our data set (**Figure 3**) so precision is very high as compare to recall and F1 score is closer to recall. **Figure 4** shows the time taken by different classifier models. Simple learners take significantly small time as compared to ensemble learners but at the cost of performance. But decision tree based classifiers (C4.5) has acceptable time as well as performance.

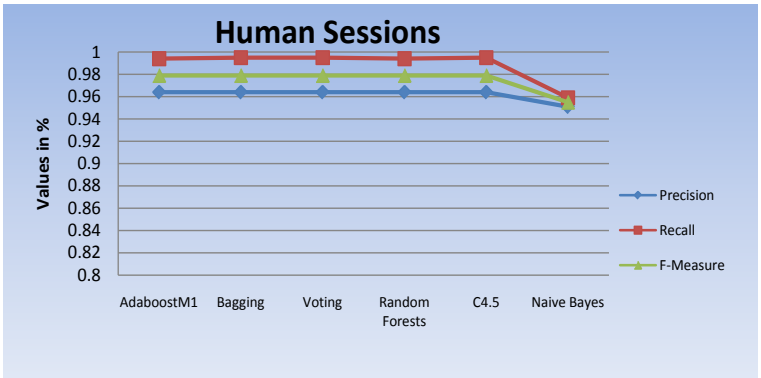

Figure 2. Comparison of measures simple vs ensemble.

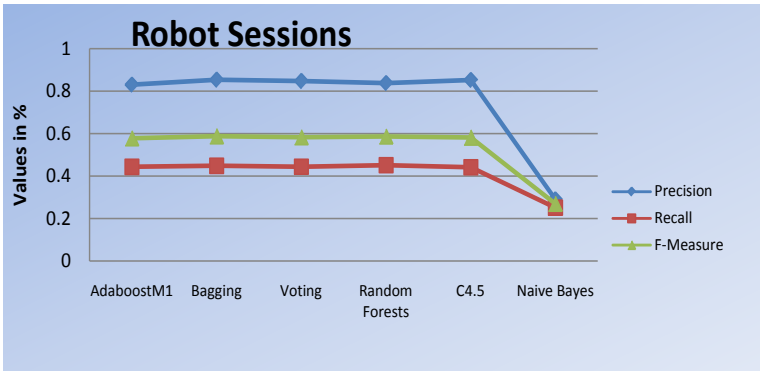

Figure 3. Comparison of measures simple vs ensemble.

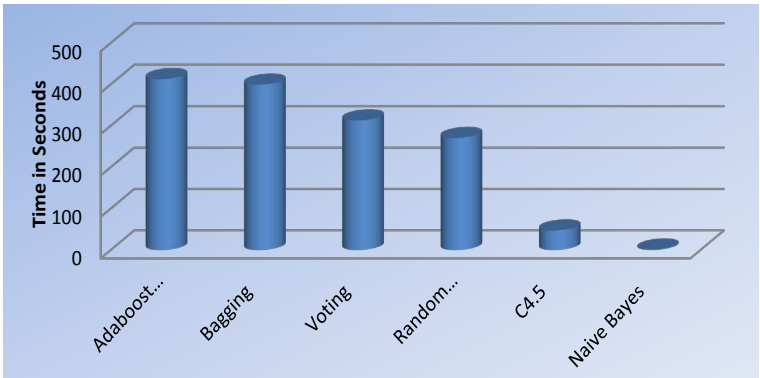

Figure 4. Comparison of model building time for classifiers.

5.2. Experiment-II

In this experiment we present the empirical evaluation on effectiveness of session length on classifiers performance using dataset with session length equal to or greater than (3, 4, 5, 6, and 7). The comparisons of the recall, precision and F1measure for the dataset with varying session length and different learners (both ensemble and simple) are shown in **Figures 5-7**, respectively. It is evident from the presented graphs that the increasing session length improves, the recall and precision and F1 measure, for all most all classifiers, except for Naïve Bays. Ensemle learners show slightly higher performance as compare to simple classifiers (same as in experiment 1) over varying length data sets. Among ensemble learners bagging performance is consistent and almost linear with data sets of increasing session length.

The results presented in the previous sections show that ensemble classifiers (boosting, bagging, and voting) achieve fairly high recall, precision and F1 measure in both experiments. Moreover, the precision for the web robots sessions (minority class) with ensemble classifiers are more than 80% in first experiment and 98% in

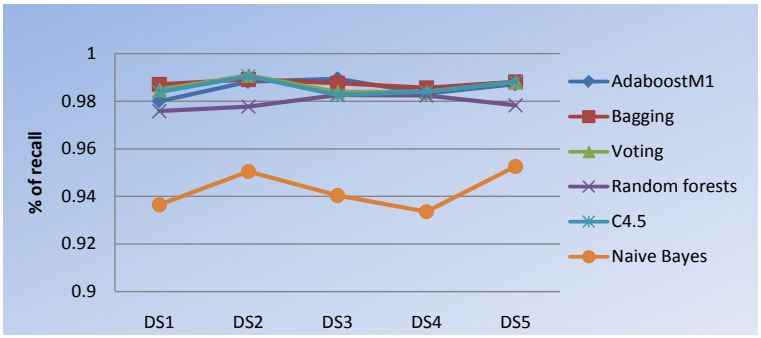

Figure 5. Precision-simple vs ensemble classifiers.

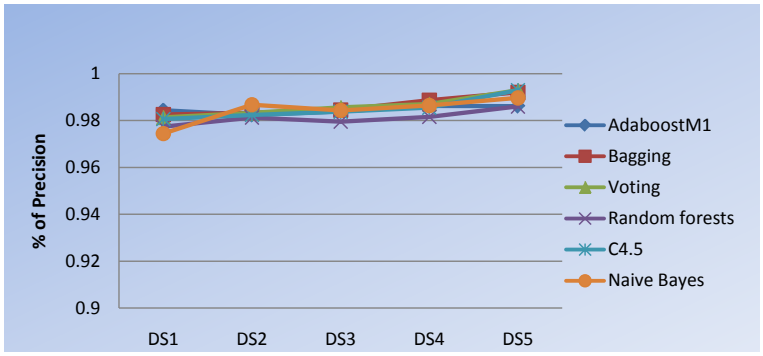

Figure 6. Recall-simple vs ensemble classifiers.

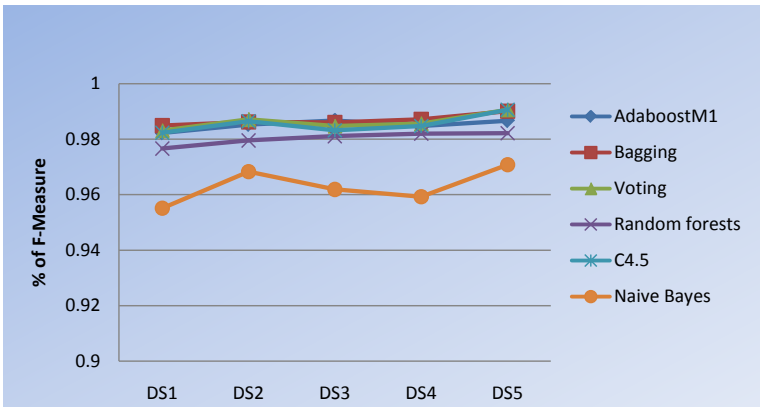

Figure 7. F1-measure-simple vs ensemble classifiers.

second Experiment. As a matter of fact, in second experiment, the F1 measure for all the algorithms which are used here are higher than the F1 measure for the same algorithms in first experiment.

6. Conclusion and Future Work

In this paper we presented the application of ensemble classifiers (boosting, bagging, and voting), for identification of web robot sessions from Web-server access logs. We evaluated the performance of these ensemble learners by using recall, precision and F1measue because these measures are well suited for this domain due to the high degree of class imbalance in the problem. The precision for the web robots sessions (minority class) with ensemble classifiers are more than 80% in first experiment and 98% in second experiment. The F1 measures for all the algorithms which are used in second experiment are higher than the F1 measure for the same algorithms in first experiment. We also presented that the increasing session length improves, the recall and precision and F1 measure, for all most all classifiers, but for Ensemble learners shows slightly higher performance compared to simple classifiers over varying length data sets. These results provide a promising direction for future work. In future we will address the issue of class imbalance with different data sampling techniques and evaluate the effect of various feature ranking methods to enhance the efficiency of automatic web robot detection.

References

[1] Tan, P.N. and Kumar, V. (2002) Discovery of Web Robot Sessions Based on Their Navigational Patterns. *Data Mining and Knowledge Discovery*, **6**, 9-35. http://dx.doi.org/10.1023/A:1013228602957

[2] Doran, D. and Gokhale, S.S. (2008) Discovering New Trends in Web Robot Traffic through Functional Classification. 2008 *Seventh IEEE International Symposium on Network Computing and Applications*, Cambridge, 10-12 July 2008, 275-278. http://dx.doi.org/10.1109/NCA.2008.47

[3] Dikaiakos, M.D., Stassopoulou, A. and Papageorgiou, L. (2005) An Investigation of Web Crawler Behavior: Characterization and Metrics. *Computer Communications*, **28**, 880-897. http://dx.doi.org/10.1016/j.comcom.2005.01.003

[4] Doran, D. and Gokhale, S.S. (2011) Web Robot Detection Techniques: Overview and Limitations. *Data Mining and Knowledge Discovery*, **22**, 183-210. http://dx.doi.org/10.1007/s10618-010-0180-z

[5] Stassopoulou, A. and Dikaiakos, M.D.M. (2009) Web Robot Detection: A Probabilistic Reasoning Approach. *Computer Networks*, **53**, 265-278. http://dx.doi.org/10.1016/j.comnet.2008.09.021

[6] Bomhardt, C. and Schmidt-Thieme, L. (2005) Web Robot Detection—Preprocessing Web Logfiles for Robot Detection. In: *New Developments in Classification and Data Analysis*, *Proceedings of the Meeting of the Classification and Data Analysis Group* (*CLADAG*) *of the Italian Statistical Society*, University of Bologna, 22-24 September 2005, 113-124.

[7] Lu, W.-Z. and Yu, S.-Z. (2006) Web Robot Detection Based on Hidden Markov Model. 2006 *International Conference on Communications, Circuits and Systems*, Vol. 3, 25-28 June 2006, Guilin, 1806-1810. http://dx.doi.org/10.1109/ICCCAS.2006.285024

[8] Guo, W.G., Ju, S.G. and Gu, Y. (2005) Web Robot Detection Techniques Based on Statistics of Their Requested URL Resources. *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design*, 24-26 May 2005, Vol. 1, 302-306.

[9] Kyoung Soo, P., Pai, V.S., Lee, K.-W. and Calo, S.B. (2006) Securing Web Service by Automatic Robot Detection. *USENIX Annual Technical Conference*, General Track, 255-260.

[10] Sisodia, D.S. and Verma, S. (2012) Web Usage Pattern Analysis through Web Logs: A Review. *Ninth International Conference on Computer Science and Software Engineering* (*JCSSE*), 2012, 49-53. http://dx.doi.org/10.1109/JCSSE.2012.6261924

[11] Myra, S., Mobasher, B., Berendt, B. and Nakagawa, M. (2003) A Framework for the Evaluation of Session Re-construction Heuristics in Web-Usage Analysis. *INFORMS Journal on Computing*, **15**, 171-190.

[12] Berendt, B., Mobasher, B., Spiliopoulou, M. and Wiltshire, J. (2001) Measuring the Accuracy of Sessionizers for Web Usage Analysis. *Workshop on Web Mining at the First SIAM International Conference on Data Mining*, 5-7 April 2001, 7-14.

[13] Useragents Database. http://www.user-agents.org/index.shtml and http://www.robotstxt.org/db.html

[14] Galar, M., Fernando, A., Barrenechea, E., Business, H. and Herrera, F. (2012) A Review on Ensembles for the Class Imbalance Problem. *IEEE Transactions on Systems, Man, and Cybernetics—Part C*: *Applications and Review*, Vol. 42.

[15] Breiman, L. (1996) Bagging Predictors. *Machine Learning*, **24**, 123-140. http://dx.doi.org/10.1007/BF00058655

[16] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. http://dx.doi.org/10.1023/A:1010933404324

[17] Schapire, R.E. (1990) The Strength of Weak Learnability. *Machine Learning*, **5**, 197-227. http://dx.doi.org/10.1007/BF00116037

[18] Freund, Y. and Schapire, R.E. (1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, **55**, 119-139. http://dx.doi.org/10.1006/jcss.1997.1504

[19] Kuncheva, L.I. (2004) Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons, Hoboken. http://dx.doi.org/10.1002/0471660264

[20] Kittler, J., *et al.* (1998) On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 226-239. http://dx.doi.org/10.1109/34.667881

[21] http://www.cs.waikato.ac.nz/ml/weka/

[22] Witten, I.H., Frank, E. and Hall, M.A. (2011) Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

[23] Fawcett, T. (2006) An Introduction to ROC Analysis. *Pattern Recognition Letters*, **27**, 861-874.

[24] Marina, S. and Lapalme, G. (2009) A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing & Management*, **45**, 427-437. http://dx.doi.org/10.1016/j.ipm.2009.03.002