

Website Visualization Software and Algorithms

Paweł Koszut

Institute of Telecommunications
Faculty of Electronics and Information Technology
Warsaw University of Technology

Abstract—Software packages which perform automated detection and visualisation of website changes serve various purposes. They are used for content monitoring in business intelligence, securing websites against illegal hacker attacks, steganalysis, censorship detection, network congestion and hardware failure detection, etc. Internet websites are subject to constant changes which reflect their natural dynamics. This characteristic behaviour can be traced automatically and analysed with appropriate software. In this paper, a software tool called 'WebGrapher' is presented which implements adaptive algorithms used effectively in cases where web page content or layout is not easily predictable (unknown web page behaviour throughout time is expected). Also, briefly discussed are other related software solutions.

I. INTRODUCTION

As Internet web pages are subject to different types of changes and modifications throughout their lifetime, classification of such changes goes not as easily as one would expect. Web page modifications result from various factors, for example, changing advertisements and banners, generated by active server-side scripts, which rotate regularly in constant time intervals. In some cases, such items can be rotated in variable time intervals adjusted in proportion to experienced HTTP traffic. Modifications on Internet web pages are also caused by their visitors. In most cases, such modifications are authorized by website administrators. This can be in case of blogging websites or other Internet portals which enable their visitors to add comments or to post messages on forums. This also takes place when web pages consist of such a dynamic content as poll and voting scripts, current time or date labels, etc. It also happens, however, that website changes may indicate unwanted actions: a hacker's attack which resulted in illegal website modifications (for example changed images), malware injections by attackers, etc. Website dynamics is also monitored for commercial reasons. In business intelligence, content monitoring can be used to react immediately in response to specified changes on competitors' websites.

Meaningful changes on interesting web pages can be detected automatically, and appropriate response actions triggered after a specified condition is met. In such cases, the trigger can also apply search-string operations, for example, verifying new product names on competitors' websites or checking and comparing prices. Automated detection of price changes is a feature which can have application in monitoring prices of traded items available at online auctions and shops. This can help win auctions by reacting automatically in response

to observed price changes. Naturally, website modifications remain unnoticed by many Internet users. Their analysis and tracking, however, can bring advantages in specific operations.

II. WEBSITE DYNAMICS

Lets start from presenting few observations. In Figure 1 and Figure 2, a sample web page which consists of a dynamic content has been presented. As we look closer into a circled content visible at the bottom of both web pages, these two samples slightly differ from each other.

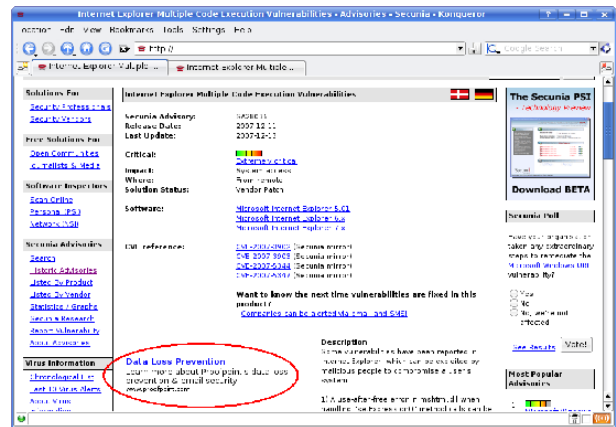


Fig. 1. The first screenshot of a dynamically changing website.

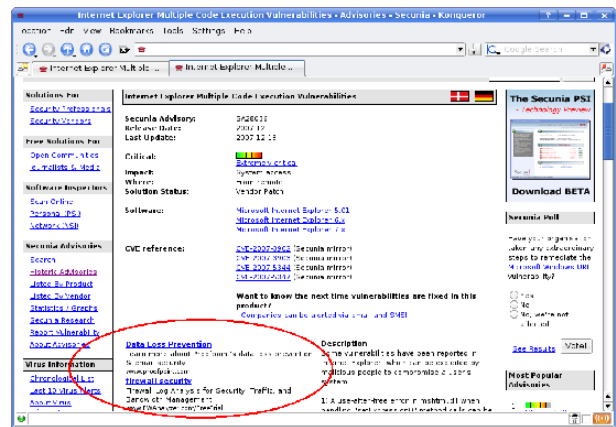


Fig. 2. The second screenshot of a dynamically changing website.

As it can be observed, circled advertisements seen at the bottom of the two web pages are not the same. This difference also means that the html files downloaded on a client's computer are different as well. The difference, however, is not radical but only minimal.

Following this observation, a graph can be constructed which illustrates such differences. During this research such graphs have been called Δ -graphs. One example of such experiments is presented below in Figure 3.

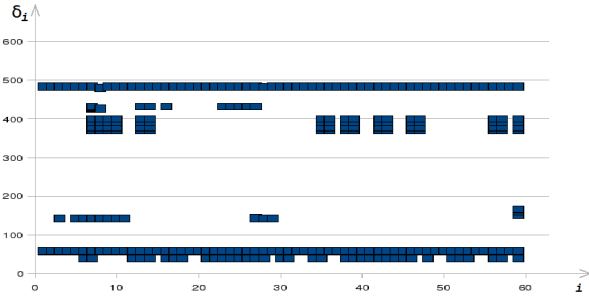


Fig. 3. The Δ -graph illustrating the sets of differences δ_i .

Horizontal axis of the Δ -graph corresponds to time, while vertical axis corresponds to web page changes within html files observed at a specific time interval. The horizontal time axis, in the presented practical implementation, illustrates $i=1..60$ consecutive html files which have been downloaded during this experiment.

On the vertical axis, these html files are characterized by changes in relation to previous files (their predecessors). The changes are illustrated by specific values which are elements of δ_i sets. The δ_i sets consist of numbers of file lines within the html files which had experienced any kind of modification.

As it can be observed, usual behaviour of web pages is that the sets of modified file lines (values on the vertical axis) remain quite predictable for the chosen web page. There are file lines which do change permanently, regularly, or occasionally, but the changes remain in foreseeable constraints.

Since the vertical axis corresponds to the numbered file lines within the downloaded html files, it is true that the maximum value within any δ_i set does not exceed the maximum number of lines within any single file of the downloaded html files.

On Δ -graphs it is easy to recognize various types of web page modifications :

- Changing banners with advertisements.
- Differences between day-time and night-time rate of change.
- Differences between rate of change on working days and on weekends.
- Changing layouts of websites.
- Adding comments or messages on forums.
- Invalid HTTP server responses (sometimes found).

Not every web page is characterised by all of the above characteristics. For example, some of web pages do change banners periodically, while other change banners on a basis of

a number of visits. In the latter case, changes illustrated on Δ -graphs are intensified during day-time hours, when visitors' activity increases. In case of periodically changed banners, where activity of visitors does not play any role, Δ -graphs illustrate constant changes regardless of the hour of the day, a day of the week.

III. WEBGRAPHER SOFTWARE

During research adaptive methods have been developed to improve tracking and visualising website changes. These methods have been later on implemented in a software called 'WebGrapher'. The software can visualise web page behaviour throughout specified period of time, while the time intervals can be exponential or adaptively changed to improve performance of a finally configured system, depending on its application. For example, in case of using the system for detecting malicious website modifications done by attackers, adaptiveness can minimize the number of so-called false-positives.

Data collected by WebGrapher is used for analysis (specifically to a user's needs) and also is used to depict graphs which will visualise web page dynamics. Such graphs are generated automatically and can serve to optimize triggering conditions configured for alerts in the system. Such alerts inform system's administrators on events of special interests.

A. Graphs obtained from real websites

Obtained by WebGrapher, in Figure 4 is an illustration of a sample web page which had been monitored for 26 days with time interval of 1 hour. This gave $26 \times 24 = 624$ html files.

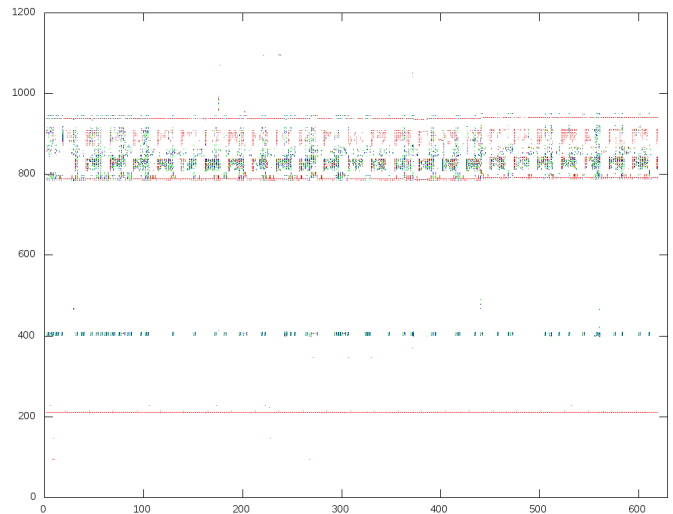


Fig. 4. The Δ -graph illustrating web page dynamics for a 24-day period.

As it can be observed, web page modifications are characterised by predictability which makes detection of changes easier. Not always, however, website changes are easily predictable. For example, when web page layout does change, the system has to adapt to a new one.

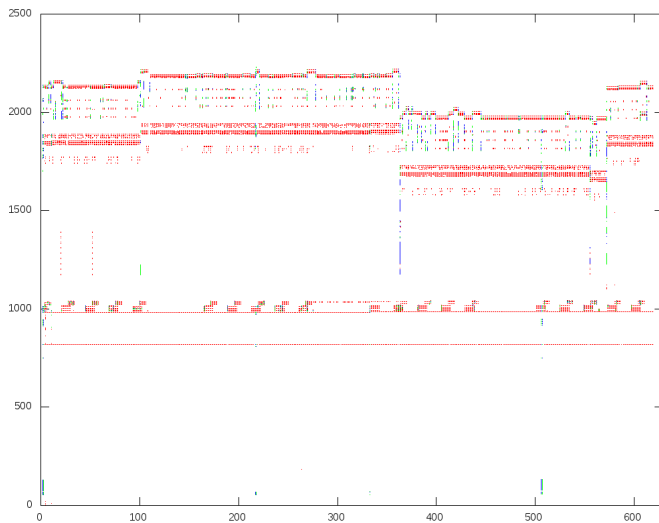


Fig. 5. Web page layout modification visible on a Δ -graph.

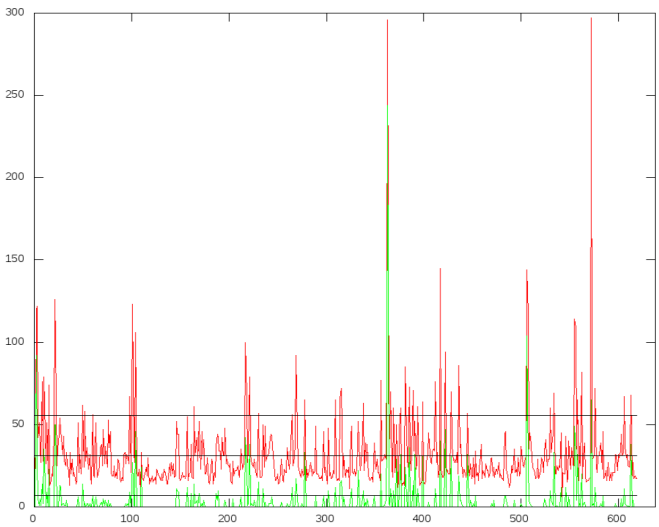


Fig. 6. Web page layout changed, illustrated by cardinality of δ_i sets.

B. Website layout change

In Figure 5, a sample layout change can be observed on a Δ -graph. As we see during a period of the experiment, monitored web page had changed significantly. We see it looking at areas on the Δ -graph which correspond to the following html files : 100th, 370th, 570th.

The web page monitored in this experiment appeared on a news website, which preserved content of the web page while changing its layout radically. Such changes are also visible on different kind of graphs, which illustrate cardinality of δ_i sets.

C. Cardinality graphs

The cardinality graph in Figure 6, displays the number of elements within δ_i sets. As it can be recognized, the significant layout changes discussed before are clearly visible on the cardinality graph. The bigger a change is, the higher values

on a cardinality graph. For the files 370th and 580th, there are two peaks corresponding the mentioned layout changes. Slightly smaller layout change is seen for the 100th file. These are visible on the previously discussed Δ -graph in Figure 5.

The δ_i cardinality graphs created by WebGrapher also include horizontal marker at the level of value 30, which is average cardinality reached for the whole html data set (624 files). This is also accompanied by additional horizontal markers at values indicating standard deviation from the computed average cardinality (30 +/- standard deviation).

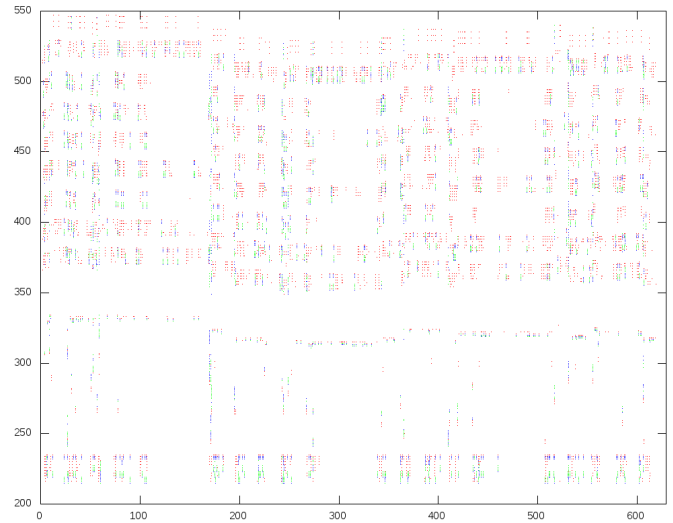


Fig. 7. The 4-week variable network traffic visible on a Δ -graph.

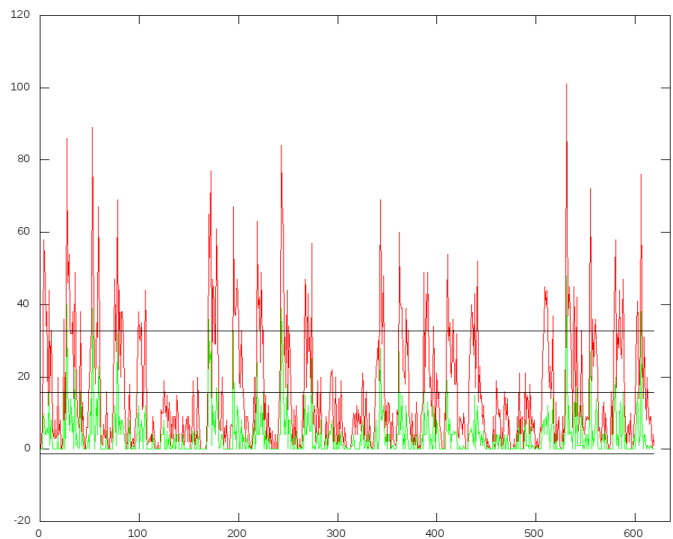


Fig. 8. The 4-week variable network traffic visible on a δ_i cardinality graph.

Talking of examples of cardinality graphs, worth observing are also graphs presented in Figure 7 and Figure 8. This pair of graphs is a Δ -graph and its corresponding cardinality graph.

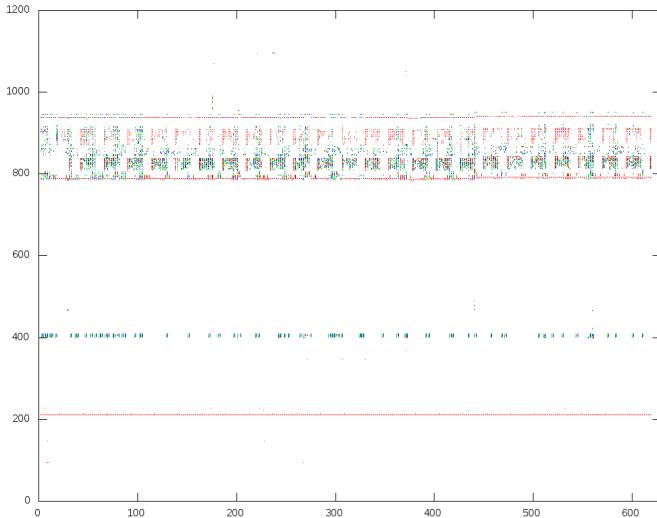


Fig. 9. Steady and predictable Δ -graph.

D. Observing HTTP traffic density

In Figure 7 and Figure 8 also another typical behaviour of web pages can be recognized. It is when detected web page modifications depend on experienced HTTP traffic. In such a case visible changes on web pages depend on a day of the week. The Δ -graph and cardinality graph, illustrated in Figures 7 and 8, show four week long observation for an example web page. The specific four weeks can be recognized on the graphs, which is a result of higher HTTP traffic experienced from Monday to Friday. This caused intensification of web page changes due to faster rotation of advertisement banners. On weekends the phenomenon diminished so the number of modifications decreased. Thus, there are visible four series of peaks which correspond to the four weeks during which the experiment was conducted. Each burst of the peaks consists of 7 peaks, 5 of which are higher due to being recorded at working days, i.e., from Monday to Friday, and 2 of them are considerably lower than others. These were recorded during weekends.

E. Number of changes per file line

Further experiments presented in Figure 9 and Figure 10 show a transformation which can be used to improve detection of some types of web page changes. Occurrences of web page changes displayed on a steady Δ -graph (Figure 9) have been summarized for each of html file lines and presented in Figure 10. The transformation shows that considerable range of file lines does not change at all. Other lines change at a predictable rate which can be used for configuration of notifications and alerts in the system. For example, when a modification is detected in “no change” zone, an alert can be triggered. Similarly, acceleration of changes in parts of html files, where changes occurred rarely before, also can be analysed.

The data set used in this graph can be treated as a measurement tool where each of html file lines is subject to analysis.

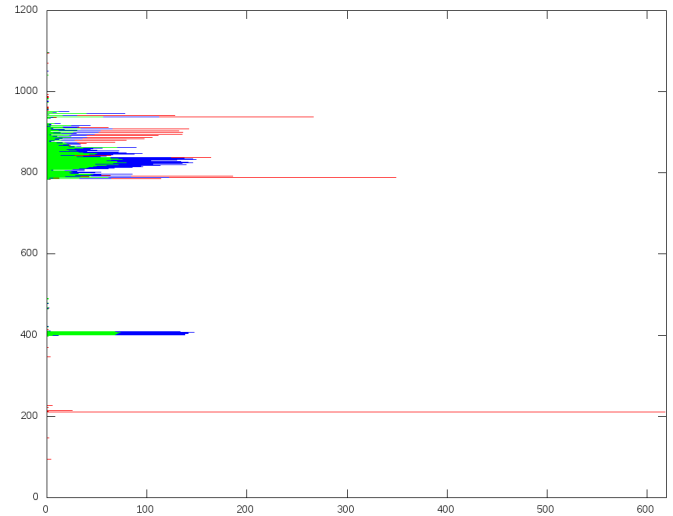


Fig. 10. Web page modifications per line number.

This indicates the rate of change per line number, where line numbers are on a vertical axis, while rate of change value (number of changes in specific period) is on a horizontal axis. This is valuable information in specific operations and can be effectively used to optimize triggering conditions for automatic alerts in the WebGrapher system.

F. HTTP invalid responses

Longer experiments revealed that visible on Δ -graphs are also failures of web services. Figure 11 illustrates a 3 and a half month long experiment, during which six HTTP error responses occurred. These are seen as vertical lines on the Δ -graph because empty (zero byte) html files, when compared with their predecessors, give δ_i sets consisting of all the possible numbers of file lines (because all of them have changed). Similar results where HTTP error responses were also found are shown in Figure 12.

G. Adding content to a web page

The graph in Figure 12 consists of additional characteristic which is its file size increasing occasionally due to forum visitors writing comments and messages on the monitored web page. This content, when added to the web page, made effects visible on the Δ -graph. This is seen as occasionally lifting upper layers of the Δ -graph.

H. Data sets

As was already said, WebGrapher uses various data sets for both analysis and visualisation. An important feature of the system is that the data sets can be used as input data for other computer programs. For example, email and sms notifications for specific anomalies detected during analysis are possible to implement because WebGrapher administrator can integrate his own additional applications with the base system, making use of available data sets. Flexible features customized for specific needs are available thanks to modular architecture of the system. This enables user-generated scripts to run

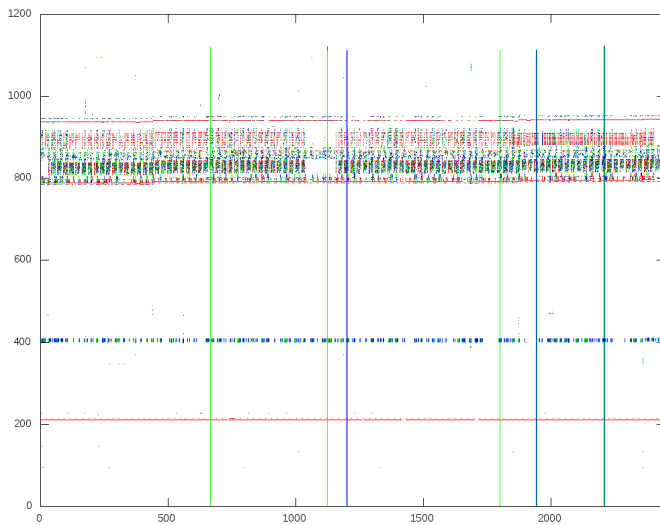


Fig. 11. Invalid HTTP responses visible on a Δ -graph.

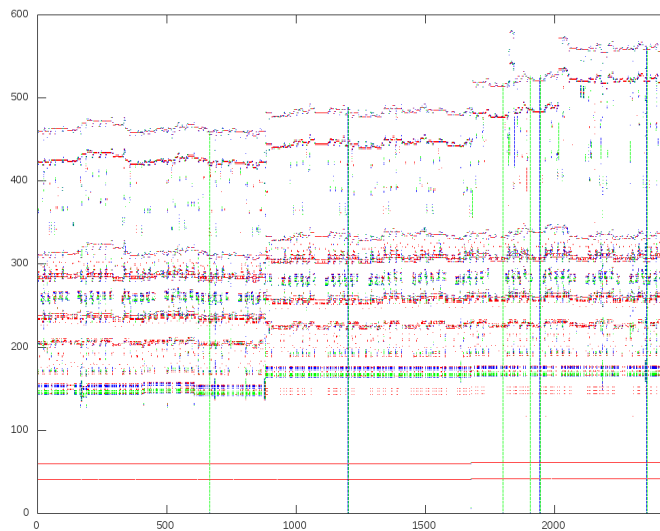


Fig. 12. Invalid HTTP responses and content inclusions seen on a Δ -graph.

any additional software to perform programmed actions. The WebGrapher data sets are available along with precomputed results presented on the graphs which were discussed till now. They were the following:

- Δ -graphs.
- Cardinality graphs.
- Web page modification per line number graph.

IV. CONCLUSION

Visualisation methods presented in this paper show innovative approach in analysing website dynamics. They explain their behaviour throughout time, giving further inspiration and opportunities to use the findings in other related works.

We can mention the following different software, algorithms and research in the area of World Wide Web: website visualisation and tracking systems[1], [2], [3], [4], [5] which are used

to create, differentiate, and analyse archived website databases, and measuring Web changes[6]. Also, there are techniques of monitoring structured data on the web[7], including XML documents[8], [9], [10]. For unstructured data techniques of identifying content blocks from web documents are used[11]. The discussed solutions have various different applications, helping web surfers to quickly identify new products, services, and automatically detect content changes on monitored web-sites.

Hopefully, this research has a potential to complement other works, giving scientists additional tool matching their needs in specific applications.

REFERENCES

- [1] Yih-Farn Chen, Eleftherios Koutsofios, *WebCiao: A Website Visualisation and Tracking System*, Proceedings of the WebNet 97 Conference, Toronto, Canada, November 1-5, 1997, AACE 1997, ISBN: 1-880094-27-4.
- [2] Fred Douglass, Thomas Ball, Yih-farn Chen, Eleftherios Koutsofios, *The AT&T Internet Difference Engine: Tracking and viewing changes on the web*, World Wide Web Journal, Vol. 1, No. 1, Baltzer Science Publishers, January, pp. 27-44, (1998), ISSN:1386-145X.
- [3] Fred Douglass, Thomas Ball, *Tracking and viewing changes on the web*, Proceedings of the 1996 annual conference on USENIX Annual Technical Conference, San Diego, California, USA, January 1996, Published by USENIX Association, 1996.
- [4] Bing Liu, Kaidi Zhao, and Lan Yi, *Visualizing Web site comparisons*, Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, USA, May 7-11, 2002, pp. 693-703, ISBN: 1-58113-449-5.
- [5] Fred Douglass, Thomas Ball, *An Internet Difference Engine and its Applications*, Proceedings of the 41st IEEE International Computer Conference Comcon'96, USA, Santa Clara, CA, 25-28 February, 1996, Published by IEEE Computer Society, pp.71-76, ISBN:0-8186-7414-8.
- [6] Fred Douglass, Anja Feldmann, Balachander Krishnamurthy, Jeffrey Mogul, *Rate of Change and other Metrics: a Live Study of the World Wide Web*, Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, California, USA, December 8-11, 1997.
- [7] Sudarshan S. Chawathe, Hector Garcia-Molina, *Meaningful Change Detection in Structured Data*, Proceedings of the ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, USA, May 13-15, 1997, pp. 26-37, ISBN: 0-89791-911-4.
- [8] Benjamin Nguyen, Serge Abiteboul, Gregory Cobena, Mihai Preda, *Monitoring XML Data on the Web*, Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp.437-448, 2001, ISBN:1-58113-332-4.
- [9] Gregory Cobena, Serge Abiteboul, Amelie Marian, *Detecting Changes in XML Documents*, Proceedings of the 18th International Conference on Data Engineering, San Jose, California, USA, 26 February - 1 March 2002, published by IEEE Computer Society Washington, DC, USA, 2002, pp. 41-52, ISBN: 0-7695-1531-2.
- [10] Yuan Wang, David J. DeWitt, Jin-Yi Cai, *X-Diff: An Effective Change Detection Algorithm for XML Documents*, Proceedings 19th International Conference on Data Engineering, Bangalore, India, March 5-8, 2003, pp. 519530, ISBN: 0-7803-7665-X, DOI: 10.1109/ICDE.2003.1260818.
- [11] Sandip Debnath, Prasenjit Mitra, C. Lee Giles, *Identifying Content Blocks from Web Documents*, Proceedings of the 15th ISMIS 2005 Conference, Lecture Notes in Computer Science, Vol. 3488/2005, Heidelberg, Germany, 2005, pp. 285-293, ISBN: 978-3-540-25878-0.