# Web Content Filtering Techniques: A Survey

V.K.T.Karthikeyan,
School of Computer Science and Engineering, Bharathidasan University, Trichy, India.
v.k.t.karthikeyan@gmail.com

**Abstract - For many, accessing the Internet is a mixed blessing; in worst case, it can create serious problems. Web Content Filtering is a firewall to block certain sites from being accessed. Content filtering and the products that offer this service can be divided into Web filtering, the screening of Web sites or pages, and e-mail filtering, the screening of e-mail for spam or other objectionable content. This paper provide a inclusive survey of major types, tasks, tools, process, an algorithm involved in the web content filtering and also suggested a new methodology to screened the text content in the WebPages and make a decision algorithm whether the webpage was allowed or banned from access.**

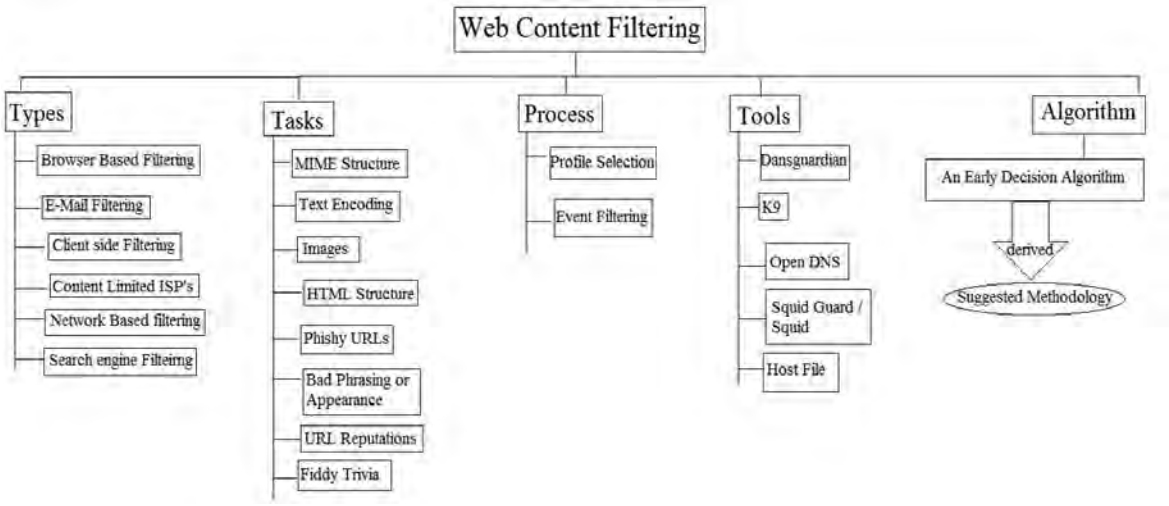
**Keywords -** Browser based filters, Text Encoding**,** Phishy URLs**,** Bad Phrasing, URL Reputation**,** Fiddly Trivia, Dans Guardian, K9, Open DNS , Squid Guard/Squid, Hosts File.

## I.INTRODUCTION

Content Filtering is new subject in the area of technology. That has to study in deep. This issue appears as consequences for the variety of media and advertisement in the internet web sites that lead to unethical and misuse of World Wide Web users**.** Massive volume of Internet content is widely accessible nowadays.

One can easily view improper content at will without access control. A modern and effective web content filtering solution scans more than the domain name. It is able to break down and analyze web traffic making it capable to accurately pinpoint portions of a web page which should not be allowed into the internal network.

Content Filtering is a firewall to block certain sites from being accessed. It is usually works by specifying character string that, if matched, indicated undesirable content that is to be screened out. Content filtering and the products that offer this service can be divided into Web filtering, the screening of Web sites or pages, and e-mail filtering, the screening of e-mail for spam or other objectionable content.




Web content filtering has following topics including Types, Tasks, Process, Tools, and Algorithm. There are six types of filtering are present they are Browser based filtering, E-mail filtering, Client-side filtering, Content limited ISP's, Network based filtering, Search engine filtering.

Web content filtering has eight types of tasks including MIME structure, Text encoding, Images, HTML structure, Phishy URLs, Bad phrasing or Appearance, URL reputations, Fiddy trivia.

Event filtering and profile selection are the two types of process involves in filtering. There are five types of tools are available in filtering they are, Dansguardian, K9, Open DNS, Squid guard / Squid, Host file.

Web content filtering has one algorithm named as An Early Decision Algorithm (described in section 7.1). From that algorithm a suggested methodology (described in section VII) is derived.

## II. TYPES

Filters can be implemented in many different ways: by a software program on a personal computer, via network infrastructure such as proxy servers that provide Internet access.

*2.1 Browser based filters*

Browser based content filtering solution is the most lightweight solution to do the content filtering, and is implemented via a third party browser extension.

*2.2 E-mail filters*

E-mail filters act on information contained in the mail body, in the mail headers such as sender and subject, and e-mail attachments to classify, accept, or reject messages. Bayesian filters, a type of statistical filter, are commonly used. Both client and server based filters are available.

*2.3 Client-side filters*

This type of filter is installed as software on each computer where filtering is required. This filter can typically be managed, disabled or uninstalled by anyone who has administrator-level privileges on the system.

*2.4 Content-limited (or filtered) ISPs*

Content-limited (or filtered) ISPs are Internet service providers that offer access to only a set portion of Internet content on an opt-in or a mandatory basis. Anyone who subscribes to this type of service is subject to restrictions. The type of filters can be used to implement government, regulatory or parental control over subscribers.

*2.5 Network-based filtering*

This type of filter is implemented at the transport layer as a transparent proxy, or at the application layer as a web proxy. Filtering software may include data loss prevention functionality to filter outbound as well as inbound information. All users are subject to the access policy defined by the institution. The filtering can be customized, so a school district's high school library can have a different filtering profile than the district's junior high school library.

*2.6 Search-engine filters*

Many search engines, such as Google and Alta Vista offer users the option of turning on a safety filter. When this safety filter is activated, it filters out the inappropriate links from all of the search results. If one knows the actual URL of a website that features sexual explicit or 18 + content, they have the ability to access it without using a search engine. Engines like Lycos, Yahoo, and Bing offer kid-oriented versions of their engines that permit only children friendly websites.

## III. TASKS

*3.1 MIME Structure*

Good email tends to be either plain text or a multipart mail consisting of two versions of the same message, one in HTML and one in plain text. Bad email often doesn't have the plain text part. Either it's missing altogether, or it's completely different (much shorter) content than the HTML part.

*3.2 Text Encoding*

Bad email often tries to hide its content from spam filters. One common way of doing this is to use base64 encoding for text where quoted-printable encoding would be appropriate. Lazy software developers sometimes base64 encode everything, as it's less work than deciding which encoding is appropriate for a message part. Doing that looks dishonest or incompetent to filters and postmasters.

*3.3 Images*

Another way bad email tries to hide its content is by misuse of images. The most obvious example of this is mail that consists of just a single huge image – sometimes that's just because it's easier for the graphic designer to do that way, but more often it's a spammer trying to hide their content from filters. Either way, it's much less likely to be delivered. Including CAN-SPAM required boilerplate (such as the postal address) purely as an image is another thing that's distinctive to bad email. Bad email hides the contact address in that way so as to avoid people being able to search based on it to track their behavior across brands and shell companies, and to stop people using it to key targeted spam filters on. Good email doesn't need to do that.

*3.4 HTML Structure*

If your email is completely unreadable with images not displayed, it's not going to be a good marketing piece in the (common) case that images aren't shown. Including appropriate ALT text for each image not only makes it look better to recipients when images are turned off, it also makes it look more legitimate to postmasters with ticketing systems that don't display images, or only show the raw HTML. It sometimes makes spam filters happier too. That's just one example of sending "good" html.

*3.5 Phishy URLs*

Bad email sent by phishers often includes links that look like <a href = http://phisher.ru/ > bank. com </a>, where the message is trying to look like legitimate email from bank.com, but it's sending readers to phisher. < ahref=http://bank.com.whatever.phisher.ru/ >bank.com</a> is an even more obvious attempt to defraud the recipient. Otherwise good email sent by naive ESPs often includes links That look like: < ahref= " http://click.esp.com? track data= xxxx & target=bank.com/">bank.com</a>. To a spam filter, that looks much the same as a typical phishing URL, and the delivery is not going to go well.

*3.6 Bad Phrasing or Appearance*

Even if 100% of your recipients desperately wait for every issue of your newsletter there are some phrases that will cause you more problems than others. "Looking spammy" is one of the worst things for your email if you need to discuss a delivery issue with a postmaster or a filter vendor – if it "looks like spam" they're much less likely to believe it's really wanted by recipients. If your newsletter is about "Moustache Rides" (real example, I'm not making this up) then you might not be able to fix the phrasing, but you should try and make the rest of the newsletter look professionally put together, as much as you can anyway.

*3.7 URL Reputation*

If two emails received "look similar" and the recipient complained about the first one, it's likely the second one will be unwanted too. But mechanically detecting similar content is complex and expensive to do, so a common trick is to "fingerprint" each email by looking for distinctive features in it, and considering messages that share a fingerprint to be similar. One of the simpler fingerprints to use is the URLs used in links in the mail, more specifically the hostnames of the links. If someone is sending bad email and you send email using the same URLs or hostnames, it's likely to be treated poorly.

*3.8 Fiddly Trivia*

There are lots of other fiddly little things that spam filters key on too. You shouldn't obsess about them too much, but it's worth being aware of the sort of things that can make a difference. Spam Assassin publish some of the rules they use. If you look at the rules, look at the scores too – a rule with a score of 0.001 isn't very relevant.

## IV. PROCESS

The first one being the *Profile Selection***,** when a collaborative user logs, all the profiles which correspond to her/his current context are selected. The second one being the *Event Filtering***,** it is performed to get only the events which match the selected profiles specifications.

## V. TOOLS

There are 5 types of tools were present in the Content-Based Filtering. They are: *Dans Guardian, K9, Open DNS, Squid Guard/Squid, Hosts File* .

*5.1 Dansguardian*

Runs on Linux, HP-UX, Solaris, FreeBSD, NetBSD, Mac-OS, Extremely configurable & allows sort of things. Blocking images, Filters unwanted ad's in your network Source machine accessing. Block the extension of the files (downloading). Controls the effect of Filters, Controls the effect of Whitelist.

*5.2 K9*

There is a limitation of working from a static databases. To overcome that, k9 introduce the Dynamic Real Time Rating (DRTR). DRTR access the Content of Websites. DRTR bans the websites if they fall into the filter categories.

*5.3 Open DNS*

Perfect solution for the Time lacking. Also for expertise to set-up & manage the server. Replace your current DNS server. Filters the connections which are send out from the source as machine. Set-up the custom filter to White list & Black list specific sites. Customize the range of filters they provide.

*5.4 Squid Guard / Squid*

Similar to Dans Guardian. Stand alone filtering tool to connect the proxys. High degree of Flexibility. Combines the filtering parameter and good change of squid guard. Natively, a Unix environmental tool.

*5.5 Host File*

Tinkering of host files with the Dans Guardian & Squid Guard. Setting up a filter in a process in great way. Essentially a mini-directory on computer IP's. Manually editing is easy but largely effectiveness. Limited to how strong the black list of downloaded items & create it

## VI. USES

Web content filtering is used to screen the web pages. whether that the page has content any unwanted content or illegal content. Corporations use as part of Internet firewall computers. In home computer, Parents use to control their children by accessing wrong websites.

## VII. ALGORITHM

*7.1 Early Decision Algorithm to Accelerate Web Content Filtering*

This work presents a simple, but effective *early decision* algorithm to accelerate the filtering from the observation that the filtering decision can be made *before* scanning the *entire* content, as soon as the content can be classified into a certain category. A fast decision is particularly important since most Web content is normally allowable and should pass the filter as soon as possible.

The philosophy behind the *early decision* algorithm is to make the filtering decision from the front partial Web content. The keyword position is normalized by the page length. The keywords in almost all Web pages tend to be distributed uniformly throughout the content or appear more in the front part according to this investigation. The Web content in a banned category starts to exhibit much more keywords than that in an allowable category since the front part. In other words, keywords from the front partial content can reveal the category of the Web content and serve as the clues to filtering.

Like the Bayesian classification, the filtering engine is trained off-line from the Web content in the banned categories. The *Bow* library and its front-end, *Rainbow* perform the training herein, extracting keywords as the features from the target categories. The keywords with the information gains larger than a threshold are selected. Stop words, such as “the”, “of” and so on, should be dropped because they help little in classification. The words inside the HTML tags are also ignored so that a malicious user cannot stuff unrelated content in the tags, particular in the front part of the Web page, to deceive the filter.

If the malicious user fills the Web text outside the tags with irrelevant content to confuse the filter, the irrelevant content will be displayed in the browser and will spoil the layout of the Web pages – a great limitation on the design of the Web pages. The score of keyword *wt* that should belong to a category *cj* is defined to be logP(*wt*|*cj*), which can be derived in the training stage. Taking the logarithm simplifies the computation of the posterior probability P(*cj*|*di*) from multiplication operations to score accumulation with independence assumption between words. The scores are accumulated while the content is scanned from the front to the end.

In the filtering stage, given *n*% of the content that has been scanned and the score *m* or less that has been accumulated, the probability that the content should belong to a category *c* is derived from

$$P(c|D(n,m)) = \frac{P(D(n,m)|c)\ P(c)}{P(D(n,m)|c) + P(D(n,m)|c')\ P(c')}$$

1. $D(n,m)$ : the event that the filter has read n% of the content and has observed the score accumulation m or less.

2. $P(c)$ : the estimated probability that category c appears in typical Web content.

3. $P(c')$ : the estimated probability that category c does not appear in typical Web content. $P(c') = 1 - P(c)$ .

4. $P(D(n,m) \mid c)$ : the estimated probability that $D(n,m)$ happens given that the content belongs to category c. The estimate of $P(D(n,m) \mid c)$ is the number of Web pages in $c$ that $D(n,m)$ happens divided by the number of Web pages in c.

5. $P(D(n,m) \mid c')$ : defined similarly as $P(D(n,m) \mid c)$ , except that $c$ is replaced with $c'$.

In the training phase, two two-dimensional indexed tables of $P(D(n, m)|ci)$ and $P(D(n, m)|ci')$ are built for each $n$ and $m$ from the training examples, where $ci$ÎC. The values of $P(ci)$ and $P(ci')$ can be estimated beforehand or dynamically tuned in a running environment by recording and analyzing actual Web content. Fig. 2 presents the *early decision* algorithm.

Two thresholds, Tbypass and Tblock, are defined to be 0.1 and 0.9 herein. *PCDi* is the estimate that the content should belong to a category *ci*. If *PCDi* is less than Tbypass for all *ci* in the list of banned categories, this means the content is unlikely to be banned and the remaining content should be bypassed. In contrast, if there exists some *ci* in the list of banned categories such that *PCDi* is larger then Tblock, this means the content is likely to belong to *ci* and should be blocked by the filter.

A minimum of the content should be scanned in the process to avoid deciding too early from only the little front part of the content, which may render the filtering result incorrect. The algorithm is

*Early bypass* ← *False;*

*Early block* ← *False;*

*n ← 0;*
Do {
Read next keyword;
// Skip stop words and the HTML tags.
*n ← the percentage of content that has been scanned;*
*m ← the accumulated score;*
If (*n > Min_Scan)*
*{*
// scanning at least Min_Scan% of document,
// Min_Scan=10 herein
For (each category *ci in the set of banned categories)*
*{*
*PDCi ← P(D(n, m)|ci) of current scanning position;*
*PDCi' ← P(D(n, m)|ci') of current scanning position;*
*PCDi ← (PDCi*P(ci))/(PDCi*P(ci)+PDCi'*P(ci'));*
}
// end of For
If (for all category *ci, PCDi < Tbypass)*
*{*
*Earlybypass:=True;*
Exit;
}
If (for some category *ci, PCDi > Tblock)*
*{*
*Earlyblock:=True;*
Exit;
}
} // End of If (*n > Min_Scan)*
while (not end of content);

## VIII.SUGGESTED METHODOLOGY

This methodology works in both a online and offline time content analysis. The philosophy behind this work is to make the filtering decision from the textual part of the Web content. Stop words, such as "a", "of", "an" and so on, should be dropped because they help little in classification. The words inside the HTML tags are also ignored so that a malicious user cannot stuff unrelated content in the tags, particular in the front part of the Web page, to deceive the filter. If the malicious user fills the Web text outside the tags with irrelevant content to confuse the filter, the irrelevant content will be displayed in the browser and will spoil the layout of the Web pages – a great limitation on the design of the Web pages.

First, count the total number of words present in the web page(A) and the total number of keywords present in the web page(B) was calculated. Next, find the categorical value(C) by calculating C=B/A. This value represents the category of the web content whether it belongs to Allowable category or Banned category. When the C value becomes greater than boundary value then the web content comes under Banned category otherwise its comes under Allowable category.

This work addresses the problem of content filtering in the web management. This methodology decides pages to either block or pass the web content as soon as the decision can be made is presented. This method is simple but effective. The same rationale behind this method can be applied to other content filtering applications as well, such as anti-spam.

This method can also be combined with more features other than keywords from the text to further increase the overall accuracy of the content filter. Besides, the filtering can be further accelerated by combining the URL-based method with the cached results. That is, by caching the URLs of the filtered Web pages, duplicate filtering on the same Web page can be avoided.

## IX.CONCLUSION

This paper presents the detail description of the web content filtering and its techniques and its types and its tasks and its process and its five types of tools and its uses and also briefly discussed about an early decision algorithm to accelerate web content filtering. Mainly this paper provides a new methodology for screening the text content in the web pages whether to allow or to ban from access. This method is derived from the basic idea of an early decision algorithm to accelerate web content filtering.

## X.ACKNOWLEDGEMENT

My sincere thanks to Dr. M Thangaraj M.Tech., Ph.D., Associate Professor, Department of Computer Science, Madurai Kamaraj University for his keen interest to encourage this work.

## XI.REFERENCES

[1] Ying-Dar Lin, Po-Ching Lin, Yuan-Cheng Lai. - An Early Decision Algorithm to Accelerate Web Content Filtering.
[2] Y.Yang and X.Liu, "A re-examination of text categorization methods", Proc. of SIGIR'99, 22nd ACM International Conference on Research and development in Information Retrieval (1999) 42-49.
[3] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Survey, vol. 34,No. 1 March (2002) 1-47.
[4] G. Cormack, "Email spam filtering: A systematic review," Foundationsand Trends in Information Retrieval, vol. 1, no. 4 , pp. 335–455, 2008.
[5] I. Androutsopoulos, G. Paliouras, and E. Michelakis, "Learning tofilter unsolicited commercial e-mail," National Centre for ScientificResearch "Demokritos", Athens, Greece, Tech. Rep. 2004/2, March
[6] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes - which naïve bayes?" in Proceedings of the 3rdInternational Conference on Email and Anti-Spam, Mountain View, CA, USA, July 2006, pp. 1–5.2004.
[7] T. Almeida, A. Yamakami, and J. Almeida, "Probabilistic anti-spam filtering with dimensionality reduction," in Proceedings of the 25th ACM Symposium On Applied Computing, Sierre, Switzerland, March 2010, pp.1–5.
[8] T. Guzella and W. Caminhas, "A review of machine learning approachesto spam filtering," Expert Systems with Applications, 2009, in press.
[9] A. Seewald, "An evaluation of naive bayes variants in content-basedlearning for spam filtering," Intelligent Data Analysis, vol. 11, no. 5, pp. 497–524, 2007.
[10] A. Bratko, G. Cormack, B. Filipic, T. Lynam, and B. Zupan, "Spam filtering using statistical data compression models," Journal of Machine Learning Research, vol. 7, pp. 2673–2698, 2006.
[11] T. Almeida, A. Yamakami, and J. Almeida, "Evaluation of approaches for dimensionality reduction applied with naive bayes anti-spam filters," in Proceedings of the 8th IEEE International Conference on Machine Learning and Applications, Miami, FL, USA, December2009, pp. 1–6.
[12] K. Schneider, "On word frequency information and negative evidence in naive bayes text classification," in Proceedings of the 4th Inter-national Conference on Advances in Natural Language Processing, Alicante, Spain, October 2004, pp. 474–485.
[13] D. Losada and L. Azzopardi, "Assessing multivariate bernoulli modelsfor information retrieval," ACM Transactions on Information Systems, vol. 26, no. 3, pp. 1–46, June 2008.
[14] J. Carpinter and R. Hunt, "Tightening the net: A review of current and next generation spam filtering tools," Computers and Security, vol. 25, no. 8, pp. 566–578, 2006.
[15] G. Cormack and T. Lynam, "Online supervised spam filter evaluation," ACM Transactions on Information Systems, vol. 25, no. 3, pp. 1–11, 2007.
[16] T. Almeida, A. Yamakami, and J. Almeida, "Filtering spams using the minimum description length principle," in Proceedings of the 25th ACM Symposium On Applied Computing, Sierre, Switzerland, March 2010, pp. 1–5
[17] Du, R.; Safavi-Naini, R.; Susilo, W.; Web filtering using text classification, The 11th IEEE International Conference on Networks, 2003. ICON2003.pages:325 - 330
[18] Weiming Hu, Ou Wu, Zhouyao Chen, Zhouyu Fu, Maybank, S., "Recognition of Pornographic Web Pages by Classifying Texts and Images", Pattern Analysis and Machine Intelligence, IEEE Transactions on, On page(s): 1019 - 1034, Volume: 29 Issue: 6, June 2007.
[19] Cao, Jiuxin , Mao, Bo and Luo, Junzhou, 'A segmentation method for web page analysis using shrinking and dividing', International Journal of Parallel, Emergent and Distributed Systems, 25: 2, 93— 104, 2010.
[20] Dontcheva, M., S. Drucker, D. Salesin, and M. F. Cohen, Changes in Webpage Structure over Time, TR2007-04-02, UW, CSE, 2007.
[21] Kim, J. K., and S. H. Lee. An empirical study of the change of Webpages. APWeb '05, 632-642, 2005.
[22] Kwon, S. H., S. H. Lee, and S. J. Kim. Effective criteria for Webpage changes. In Proceedings of APWeb '06, 837-842, 2006.
[23] Arasu, A. and Garcia-Molina, H (2003). Extracting Structured Data from Web Pages. SIGMOD-03.
[24] Chen, Z., O. Wu, M. Zhu, and W. Hu (2006) A novel web page filtering system by combining texts and images. In WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, pp. 732–735. IEEE Computer Society.
[25] White Paper, Meeting the challenges of Web Content Filtering - March 2007 .