28-9-2003

Web filtering using text classification

R. Du
University of Wollongong, rongbo@uow.edu.au

R. Safavi-Naini
University of Wollongong, rei@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Web filtering using text classification

Web Filtering Using Text Classification*

Rongbo Du, Reihaneh Safavi-Naini and Willy Susilo
Centre for Communication Security
School of Information Technology and Computer Science
University of Wollongong, Australia
Email: {rd12, rei, wsusilo}@uow.edu.au

Abstract- **Web filtering is used to prevent access to undesirable Web pages. In this paper we review a number of existing approaches and point out the shortcomings of each. We then propose a Web filtering system that uses text classification approach to classify Web pages into desirable and undesirable ones, and propose a text classification algorithm that is well suited to this application.**

I INTRODUCTION

Web filtering uses screening of Web requests and analysis of the contents of the received Web pages to block undesired Web pages. Web filtering has the following major applications.

(i) Protection against inappropriate content: Internet is becoming an important source of information. However it is also host to pornographic, violent and other contents that are inappropriate for most viewers. Web filtering can be used to block access to pages that are against a defined policy.

(ii) Preventing misuse of the network: Here the main aim is to prevent misuse of the resources of an organisation. A common problem in many organisations that provide Internet access for employees is that the network connection could be used for applications such as chat, network games, and downloading streaming video and audio content. Such misuses decrease productivity and impose unnecessary load on the entire network, impeding legitimate activities. In some cases such as downloading illegal media files or software, there might be legal implications not only for the employees, but also for the employees' organisation.

Current implementations of Web filtering uses techniques such as blacklisting or whitelisting, keyword searching and rating systems. Blacklists and whitelists are costly to generate and maintain. Keyword searching is prone to spelling mistakes that can be used to bypass this protection. Rating systems in general do not provide a reliable source of information.

In this paper we propose a text classification approach to determining the pages that must be blocked. The "forbidden" class is defined in terms of a set of documents, each a sample forbidden Web page. For a candidate Web page, the similarity with this defining set is measured and if greater than a predetermined threshold, it is considered belonging to the class. A similar approach has been proposed by Lee et al [7], where artificial neural networks were trained to identify members of the forbidden class. However, the proposed system is computationally expensive and the set of negative (not belonging to the forbidden class) documents for training could be massive. In our system, each document is represented by a vector of word probabilities and similarity of two documents is measured using 'cosine' of the angle between the two associated vectors. We define the 'cosine' in terms of the relative inner product of the two vectors. This results in a very efficient (computationally) algorithm that can be incorporated into firewalls without major slow down of the network traffic.

First we briefly review existing Web filtering systems and text classification algorithms. We propose a Web filtering system using text classification and give a new classification algorithm that is well suited for this application. Details of our implementation are given. We also discuss filtering non-text data such as images and propose a simple method of filtering forbidden images. We also show limitation of Web filtering by describing a simple attack on such systems. Finally we provide some further topics.

II RELATED WORK

Previous Web filtering approaches include the following.

Blacklists and Whitelists. Blacklists and whitelists are lists of Web sites that must be blocked or allowed, respectively. Blacklists are usually created by examining Web sites manually and deciding whether a site can be classified as a member of a forbidden class, such as "Nudity" and "Violence". Sites can also be automatically included in blacklists if their domain name contains keywords like "sex" or "xxx". In whitelisting, a list of permissible sites is generated and anything else is blocked. The main problem with both these lists is that because new sites continually emerge, it is hard to construct and maintain complete and up-to-date lists.

Keyword Blocking. In this approach a list of keywords is used to identify undesirable Web pages. If a page contains a certain number of forbidden keywords, it is considered undesirable. The problem with this method is that the meanings of words depend on the context. For example, sites about breast cancer research could be blocked because of the occurrence of the word "breast" that is used as a keyword for "pornography class". A second problem is that the system is easily defeated using words intentionally or unintentionally mis-spelled. For

*This work was partially supported by Smart Internet Technology Cooperative Research Centre, Australia.

0-7803-7788-5/03/$17.00 ©2003 IEEE.

example, a malicious site can replace the word "pornographic" with "pornogaphic" to thwart filtering systems. Such replacement will have little effect on the readability of the page by human users but would make it significantly more difficult for filtering systems to correctly find the original keyword.

Rating Systems. Rating systems such as PICS (Platform for Internet Content Selection) [14] can produce rating for Web sites. PICS specification associates metadata to Web pages. There are two approaches to rating of sites. In *Self-Rating*, Web page publishers generate their own rating information. In *Third-Party Rating*, an independent third party is used to evaluate Web sites and publish the results. This information can be used for Web filtering purposes. The problem with rating systems is that rating is not compulsory and so is not always available. Moreover, because of the possibility of self rating, ratings are not always reliable and accurate.

Almost all existing filtering software use blacklists and whitelists, while some also provide rating and keyword option. Most of the systems [10, 11, 15] are stand-alone applications or plug-ins that reside on end users' terminals. (Plug-ins are programs that are installed as part of the Internet browsers.) Performance of a filtering system can be measured in terms of *blocking rate* which is the percentage of the correctly blocked Web pages, and *overblocking rate* which is the percentage of legitimate pages that are blocked. The NetProtect project evaluated 50 commercially available filtering systems using 2,794 URLs with pornographic content and 1,655 URLs with normal content [12]. Their results reproduced in Table I show that the accuracy of existing systems is far from satisfactory.

Table I: NetProtect's Evaluation for filtering tools

Filtering Tools	Blocking Effectiveness	Overblocking Rate
BizGuard	55 %	10 %
Cyber Patrol	52 %	2 %
CYBER sitter	46 %	3 %
Cyber Snoop	65 %	23 %
Internet Watcher 2000	30 %	0 %
Net Nanny	20 %	5 %
Norton Internet Security	45 %	6 %
Optenet	79 %	25 %
SurfMonkey	65 %	11 %
X-Stop	65 %	4 %

II-A Implementing Web Filtering

There are two main methods of implementing Web filters: implementing as a separate filter on each end-computer, and implementing as part of a firewall that controls the traffic of the network. Compared to end-computer based filtering, filtering at firewalls has the advantage of making it harder for malicious users to disable or circumvent the filtering process. Web filtering programs as part of the firewall execute on the bastion host which is not accessible to normal users. Another advantage is that since all network traffic to external networks passes through the firewall, it is hard to escape the filter through direct communication with outside hosts. A third advantage of Web filtering at firewalls is the ease of maintenance and update of the system. Finally, this architecture is more suitable for end-computers with weak computational power or limited storage space that are behind a firewall. There are also some disadvantages in filtering at firewalls. Firstly, this approach slows down the entire network because all incoming traffic passes through the firewall. Another disadvantage is that if end-to-end encryption is used, the traffic through the firewall is encrypted and Web filtering at the firewall cannot be performed. Filtering at end-computers complements Web filtering at firewalls and its advantages and disadvantages can be directly concluded from the above discussion.

II-B Text Classification

Automatic text classification has been of growing importance because of the rapid increase in generation of text documents in recent years. Automated text classification is a supervised learning task that assigns pre-defined category labels to new documents using comparison with a training set of labelled documents [19]. Traditional automatic text classification systems are used for simple texts and so their application to Web pages that are hypertexts need careful considerations. The main approaches to text classification are: Naive Bayes (NB) [8], K-Nearest Neighbor (KNN) [19], Decision Tree (DTree) [9], Support Vector Machines (SVM) [6] and Neural Network (NNet) [18].

Naive Bayes (NB) classifiers are widely used because of their simplicity and computational efficiency. NB uses relative frequencies of words in a document as words probabilities and uses these probabilities to assign a category to the document. NB assumes that the conditional probability of a word w, given a category C, denoted by $P(w|C)$, is independent for different values of w.

K-Nearest Neighbor (KNN) is a statistical approach which is among the most accurate methods of classifying documents. Given a document, KNN selects k most similar documents from the training set and uses the categories of these documents to determine categories of the document being classified. Documents are represented by vectors of words and the similarity between two documents is measured using Euclidean distance or other functions between these vectors [19].

Decision Tree is a machine learning approach to automatic induction of classification trees based on training data [9]. Each internal node of the decision tree is associated with a test on an attribute and outgoing branches of the node correspond to the results of the test. A leaf is associated with a category. Classification of a document starts from the root node and then internal nodes are successively visited until a leaf is reached. At each node the test associated with the node is performed to determine the next node. The document category is the category of the final leaf [13].

Support Vector Machines (SVM) uses decision surfaces to divide data points into classes [16]. SVM is also applied to text classification [6]. In its simplest form, training documents are represented as vectors and the algorithm determines hyperplanes that separate different classes of training documents. Test documents are classified according to their positions with respect to the hyperplanes.

Classification of hypertext data. Yang et al [20] used a common data set to compare the effectiveness of NB and KNN

algorithms for classifying Web pages. They studied the usefulness of hyperlinks, content of linked documents, and meta data in classification, and found that meta data can increase the accuracy of classification by a large factor [4].

Text classification for filtering. Lee et al [7] applied artificial neural network to filter pornographic pages. They used a collection of pornographic and non-pornographic pages to train artificial neural network which could be used to decide if a given Web page is pornographic. The method requires high computation and hence is unsuitable for real-time application.

III A NEW PROPOSAL FOR WEB FILTERING

Blacklists and whitelists are hard to generate and maintain. Also filtering based on naive keyword-matching can be easily circumvented by deliberate mis-spelling of keywords and techniques to overcome this problem result in high computation and increased number of false positives. Finally, rating systems do not provide reliable information.

We propose a new Web filtering method based on text classification. We use samples of forbidden Web pages to characterise the class of Web pages that must be blocked. A Web page that is 'close', or 'similar', to members of this class is blocked and those that are 'dissimilar' will be allowed.

In applying text classification algorithms a number of points must be taken into account. Firstly, classification for Web filtering is a one-class classification where the result of classification is either, an 'allowed' or a 'blocked' page. The classification determines if a Web page belongs to a forbidden class, for example is a pornographic page or not. Most traditional text classification systems require *positive*, that is documents having the same characteristics of a class, and *negative*, that is documents that do not have characteristics of that class. In the case of classification of Web pages, it is not easy to provide a representative sample of the *negative* class because of the variety of documents in this class.

Our proposed classification method requires only *positive* training documents and so removes the problem of constructing and maintaining a complete and balanced set of *negative* documents. Moreover, in traditional text classification problem, documents that need to be classified are considered independent and so classification of one document does not provide useful information about classification of other documents. In Web filtering, Web pages that can be reached through hyperlinks in the document also provide useful information for the classification of the document. In particular in marginal cases where the page content cannot give a clear classification of the document, using hyperlinks to find pages that are considered similar to the page under investigation, can be very useful.

III-A The Algorithm

Each document is represented by a vector of frequencies of words. The length of the vectors will be N and so only the frequencies of N most frequent words will be kept. The similarity between two documents is measured in terms of the 'cosine' of the 'angle' between the two vectors with more similar documents having the smaller angle and so larger 'cosine' value, and the less similar ones having larger angle and so smaller 'cosine' values. The *cosine* value between two vectors can be calculated as

$$cos(X, Y) = \frac{\sum X_i Y_i}{\sqrt{\sum X_i^2 \sum Y_i^2}}$$

where X and Y are the two documents' vectors.

The training set $\mathcal{T}$ consists of sample Web pages with forbidden content. To classify a page, its similarity to $\mathcal{T}$ is measured and if it is above a threshold, it is considered belonging to the forbidden class. To determine the threshold, another data set $\mathcal{T}'$ that consists of samples with forbidden content and samples of allowed Web pages is constructed. Then for a range of threshold candidates, we use each candidate to classify members of $\mathcal{T}'$ and choose the threshold τ that correctly classifies most members of $\mathcal{T}'$(see following for details). To calculate the *similarity coefficient* of a page P to the class defined by $\mathcal{T}$, P's similarity with every training document in $\mathcal{T}$ is found and then the average of the n% highest similarity values is used as P's *similarity coefficient* to $\mathcal{T}$. Here n is a number that depends on the number of "sub-groups" in $\mathcal{T}$. For example, "sex" category may consist of two sub-groups: "erotic stories" and "pornographic galleries". Since a document belongs to one sub-group may not necessarily be similar to the other sub-group, for documents belong to one particular sub-group, averaging the top 50% similarity values will result in a higher *similarity coefficient* to the "sex" category than averaging all similarity values.

If P's *similarity coefficient* to $\mathcal{T}$ is less than the threshold, the hyper-links within P are examined and the *similarity coefficient* of the pages that the link is pointing to is found. This will be done for r links. If in most cases the link is pointing to a page similar to the forbidden category then P is also classified as forbidden.

The steps of the algorithm can be summarised as follows.

1. Enter the set $\mathcal{T}$ of training documents where each document belongs to the forbidden class. This set is chosen in the initialisation phase and is updated regularly.

   For a document $T \in \mathcal{T}$ an associated vector v_T of relative frequencies of words, using the following steps, will be constructed.

   (a) Remove common words such as "the", "and" and "for". (This is because these words appear in all documents and will not contribute to classification of documents.)

   (b) Ignore low frequency words. (These words are unlikely to be indicative of the type of the document.)

   (c) Ignore words shorter than 2 letters. (This is to avoid the count of words such as "a", "to", "of" and"in", and also special symbols such as "@", "?", etc.)

   During initialisation phase, the vectors corresponding to all training documents are found.

2. Find the threshold τ to be used in deciding if a document belongs to the forbidden class. Using higher thresholds

will result in pages that do belong to the forbidden class to be missed, and using lower thresholds will result in pages which do not belong to the forbidden class to be wrongly blocked. These two cases result in systems' *false negatives* and *false positives*, respectively. After vectors of all training documents are found, we use another data set $\mathcal{T}'$ that consists of samples inside and outside the forbidden class and measure the *similarity coefficient* of each element of $\mathcal{T}'$ to $\mathcal{T}$(see following step for how to calculate *similarity coefficient*). We use a range of threshold values (from 0 to 1) and use each to classify members of $\mathcal{T}'$. Let u_{τ_i} denote the percentage of documents in $\mathcal{T}'$ that are correctly classified using threshold τ_i. We choose the threshold $\tau = \tau_j$ that has the highest u_{τ_j} for the system.

3. For a Web page P, the system finds the *similarity coefficient* σ_P as follows.

   (a) Find similarity of P with every member of the training set. That is, find $cos(v_P, v_X)$, $X \in \mathcal{T}$.

   (b) Find $\mathcal{S}$, the set of the $n\%$ highest similarity values.

   (c) Class coefficient of P, σ_P, is the average of the $n\%$ highest similarity values. That is,

$$\frac{\sum_{v \in \mathcal{S}} v}{|\mathcal{T}| \times n\%}$$

   The system compares σ_P with the chosen threshold τ. If $\sigma_P \geq \tau$, the page is blocked, otherwise the system considers the r randomly chosen hyperlinks $\ell_1, \cdots \ell_r$ in the page, and for each measures the *similarity coefficient* of the document P_{ℓ_i} that is pointed to by ℓ_i. If the majority of the *class coefficients* are above the threshold τ, the page is blocked, otherwise allowed.

Using HTML Tags to Improve Accuracy of Web Filtering. HTML tags contain instructions to browsers. For example, tag *title* specifies the title of a Web page and tag *h1* instructs the browser to display text within the tag as headings. HTML tags may be used to improve Web filtering. The tags are used to measure relative significance of words. For example, the words within the tag *title* will be counted with a weight of 3. Our preliminary experiments show that using the tag *title* with a higher weight (e.g. 3) improves correct classification while other tags such as *h1* and *h2* are not so effective.

IV WEB FILTERING AT FIREWALL

Since our Web filtering approach is based on the content of Web pages that may extend over a number of IP packets, filtering must be performed after the page is reconstructed in a single file. Figure 1 shows the location of Web filtering with respect to the access control of an already existing firewall. Web filtering happens after access allowed. Only network traffic that is permitted by the firewall is checked for possible undesired content. A text-based Web filtering component processes contents in text form and so filtering methods such as naive keyword-matching method, or other sophisticated methods of text analysis can be used.

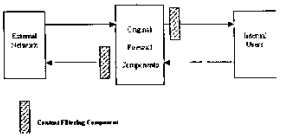

Figure 1: Architecture of A Web filtering Firewall

V FILTERING NON-TEXT FILES

Web pages are mostly in text form and so text-based analysis will be effective in most cases. Another important form of representing information is images. This form can be used to thwart text-based filtering. There are a number of ways of communicating forbidden information through images.

The forbidden information is in visual form with no written text attached to it. Prime examples of such information is pornographic information. If no textual information accompanies a forbidden image, the only way to filter the information is using image analysis systems. There are a number of image analysis systems, such as *Fleck-Forsyth-Bregler* system [3] and *WIPE* [17] that classify images as pornographic or non-pornographic. In general image filtering systems can produce good results, but they are not suitable for real-time filtering systems. We also note that all such systems are aimed at pornographic images and it is not clear how to extend them to other undesirable categories (such as violence) and how good the results will be.

A second way of image representation for transmitting forbidden information is by converting text to image. Such conversion will result in a non-text representation of the information and will make the text-based algorithms ineffective. Applying character recognition algorithms to non-text contents could recognise enough characters to make the text-based algorithms useful. In general because text version of the information is much more compact, conversion to images will not be widely used.

Finally data hiding techniques such as watermarking can be used to communicate forbidden information. In general it is very difficult, if possible at all, to detect and protect against this kind of communication. Although in some cases it might be possible to detect presence of hidden data in a "cover file", for example an image file, there is no general method of protecting against such embedded data.

As seen above image analysis techniques in general are unsuitable in firewall filtering. An alternative is to use meta data in HTML documents to provide some initial filtering. In particular, we may use the *alt* attribute of HTML *img* tag. The *alt* text of an image will be displayed in one of the following cases: 1) the image cannot be displayed, either because the viewer has disabled the "display image" option of the browser or because the image is not currently available, 2) the viewer is using a text-based browser such as *lynx* and so the image cannot be displayed, or 3) the viewer's mouse is positioned over the image. The *alt* attribute provides alternative information to the viewer. Because most *alt* texts are related to their corresponding images, the *alt* texts can be used to obtain some

information about the images.

The advantage of this method is its simplicity and high speed because the filtering system will remain text-based. Implementation of this system is straightforward and requires few modifications to the HTML parsing component of the original filtering system. The disadvantage is its limited effectiveness because not all *alt* texts provide meaningful information about the images.

VI IMPLEMENTATION

We implemented the proposed Web filtering module in Java 1.4. We collected Web pages from various sites and used them to test our filtering system. The experimental results have been very encouraging.

The Web filtering component was incorporated into a firewall that has been implemented using the firewall toolkit (FWTK) [5]. The firewall toolkit is a set of components that can be used to create an application-level firewall system. The toolkit is chosen for this implementation because its source code is publicly available.

In our experiment, the forbidden pages belong to the category of Adult content. We collected 487 URLs from the Adult category of Google [1]. Web sites in this category have been reviewed and classified as containing adult contents by human editors. We obtained the list of top 500 adult Web sites by searching with the keyword *porn*. From the 500 returned URLs, 13 were invalid or unavailable. We used the remaining 487 URLs as the training set $\mathcal{T}$. To determine the accuracy of our system, a test set $\mathcal{T}'$ was constructed, which consists of URLs with Adult contents ($\mathcal{T}_1'$) and URLs without Adult contents ($\mathcal{T}_2'$). For all the URLs (385 when the data were collected) under the Adult category of Yahoo [2], we excluded the URLs that were already in $\mathcal{T}$ or invalid and used the remaining 329 URLs as $\mathcal{T}_1'$. The filtering system was expected to recognise URLs belong to $\mathcal{T}_1'$ as containing adult contents. Also, URLs that do not contain Adult contents were collected from 10 top directories of Google. The categories were: *arts, business, science, computers, news, shopping, games, society, health,* and *sports*. For each category, we obtained a list by searching with the keyword same as the category name (For example, keyword *arts* for *arts* category.). For each list, we selected the top 20 URLs returned, the middle 20 URLs and the bottom 20 URLs. For the selected 600 URLs, we excluded invalid or unavailable URLs and obtained 587 URLs. We used the 587 URLs as $\mathcal{T}_2'$. Since $\mathcal{T}_1'$ and $\mathcal{T}_2'$ are disjoint, therefore, $|\mathcal{T}_1'| + |\mathcal{T}_2'| = |\mathcal{T}'|$, where $|E|$ denotes the size of set E. In total, 1,403 URLs were collected and used for the experiment.

VI-A *Analysis of Experiment Results*

The experiment was run multiple times. In each run, we selected 10% of $\mathcal{T}'$, u, as the test data for the run. Using the same training data $\mathcal{T}$, for each test document in u, the filtering program calculates the *similarity coefficient* to $\mathcal{T}$. Then with different thresholds, the filtering program calculates the corresponding blocking rate and over-blocking rate. When the multiple runs finish, we calculate the distribution of blocking rate and over-blocking rate and the 95% confidence intervals with different thresholds.

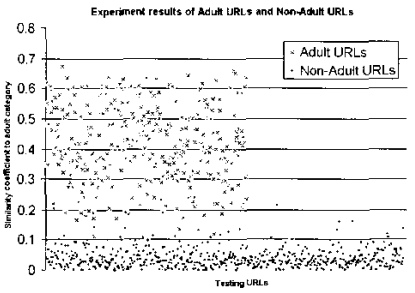

Figure 2: Experiment results of Adult URLs and Non-Adult URLs

Figure 2 shows the *similarity coefficient* to the Adult category calculated by the filtering system for all the members of $\mathcal{T}'$. The Y-axis is the *similarity coefficient* to $\mathcal{T}$ and its range is between 0 and 1. The X-axis is the set of URLs in $\mathcal{T}'$. A test URL is given high *similarity coefficient* if the filtering system considers the URL contain adult contents. A test URL is given low *similarity coefficient* if the filtering system considers it does not contain adult content. It can be observed that our filtering system gives Web sites that do contain adult content high *similarity coefficient* to $\mathcal{T}$ (most are higher than 0.2), while Web sites free of adult content have low *similarity coefficient* (generally less than 0.2). We ran experiment 100 times (each time using 10% randomly chosen URLs from $\mathcal{T}'$) and obtained the following results for different threshold values:

Table II: Results for Experiments (100 runs)

Blocking Rate (Avg)	*Over-Blocking Rate(Avg)*	*Threshold*	*95% Confidence Interval of BlockingRate*	*95% Confidence Interval of Over-BlockingRate*
99.35%	4.09%	0.10	99.21% to 99.48%	3.75% to 4.43%
98.99%	2.03%	0.12	98.84% to 99.15%	1.79% to 2.28%
98.83%	0.92%	0.14	98.62% to 98.98%	0.74% to 1.09%
98.80%	0.59%	0.16	98.62% to 98.98%	0.46% to 0.71%
97.41%	0.48%	0.18	97.13% to 97.68%	0.36% to 0.59%
95.34%	0.34%	0.20	94.98% to 95.71%	0.24% to 0.44%
93.67%	0%	0.22	93.27% to 94.07%	0% to 0%
...	...	...	...	...

Total number of adult URLs tested: 329
Total number of non-adult URLs tested: 587
Total number of adult URLs used for training: 487

$$BlockingRate = \frac{\text{Total number of adult URLs blocked}}{\text{Total number of adult URLs tested}}$$

$$Over-BlockingRate = \frac{\text{Total number of Non-adult URLs blocked}}{\text{Total number of Non-adult URLs tested}}$$

The accuracy of the filtering system is shown to be satisfactory.

We identified and analysed Web pages that resulted in false positives or false negatives. Most such pages contain only small amount of text (typically less than 20 words) and since our filtering systems is text-based it produced incorrect result.

VI-B *Comparison with Existing Systems*

The NetProtect project evaluated the filtering efficiency and over-blocking rates of existing filtering systems, using 2,794 URLs with pornographic content and 1,655 URLs with normal content to test 50 commercial filtering systems [12]. Because

of the similarity of the goal of the NetProtect experiment to our experiment, we can compare our results to theirs (see table I). We could not use the same set of URLs in our experiments because NetProtect did not publish details of their tests. The results clearly show that our filtering system is more accurate than existing filtering systems for adult contents. It should also be pointed out that our system does not use any blacklist or whitelist and therefore avoids the cost of manually classifying and maintaining such lists.

Lee et al [7] applied artificial neural network to filter pornographic Web pages. Compared with their results, our system is more accurate and faster. Also, a major disadvantage of their system is that some Web pages cannot be classified and are reported as "unascertained". In their experiments, around 5% to 12% Web pages are reported "unascertained". In practise, a filtering system has to make a decision to block or allow a page and an "unascertained" result must be converted to block or allow. The disadvantage is that allowing all "unascertained" Web pages decreases blocking rate, while blocking all "unascertained" Web pages increases overblocking rate. Our filtering system always returns a definite result.

VII POSSIBLE ATTACKS ON WEB FILTERING

Web filtering firewalls are not effective if network traffic is encrypted and cannot be decrypted by the firewall, or more generally, if the traffic is in a form that the firewall cannot interpret correctly. A simple attack would be to use Java applets and XML (Extensible Markup Language). Java Applet is a program written in Java language that can be sent to the user's machine to be executed (with restrictions). XML is used to describe, store and exchange data. Suppose an outsider wants to send pornographic content to users behind a Web filtering firewall. He first encrypts the pornographic content, puts the key and encrypted content in XML file, and then creates a Java applet for decrypting the XML file. When a user accesses the Web site, the XML file and Java applet are sent through the Web filtering firewall without being blocked because the firewall cannot decrypt the encrypted content and is not even aware that the XML file contains encrypted content. At user's machine, the encrypted content will be decrypted by the applet and displayed. We have demonstrated this attack. Although we can block all applets by removing the *applet* tags in Web pages, other legitimate applets will also be blocked.

Another possible attack is to insert some benign words or links into forbidden pages in order to confuse the Web filtering. This attack may work if the inserted text is a considerable portion of the page. In this case not only the Web filtering system be confused but the page viewers will also be disrupted and the search engines be confused. Reduced readability of pages and reduced ranking for search engines will make this approach of less use.

VIII CONCLUSIONS AND FUTURE WORKS

To obtain satisfactory Web filtering results, we proposed and implemented a new Web filtering system that uses text classification. Further interesting topics include: (i) How to correctly and effectively filter Web pages with multi-media contents, such as flash. (ii) How to combine filtering at firewall and filtering at terminals to come up with a better filtering system. (iii) How to deal with encrypted network traffic.

References

[1] http://directory.google.com/top/adult/.

[2] http://dir.yahoo.com/business_and_economy/shopping_and_services/sex/directories/.

[3] Margaret M. Fleck, David A. Forsyth, and Chris Bregler. Finding naked people. In *ECCV (2)*, pages 593–602, 1996.

[4] Rayid Ghani, Seán Slattery, and Yiming Yang. Hypertext categorization using hyperlink patterns and meta data. In *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 178–185, 2001.

[5] Trusted Information Systems Inc. TIS Firewall Toolkit.

[6] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning.*

[7] Pui Y. Lee, Siu C. Hui, and Alvis Cheuk M. Fong. Neural Networks for Web Content Filtering. *Intelligent Systems*, 17(5):48–57, Sep/Oct, 2002.

[8] David D. Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval.*

[9] T. Mitchell. *Machine Learning.* McGraw Hill, 1996.

[10] Net Nanny. *At http://www.netnanny.com/*, online.

[11] Cyber Patrol. *At http://www.cyberpatrol.com/*, online.

[12] NetProtect Project. Report on currently available cots filtering tools. Technical report, 2001.

[13] Rajeev Rastogi and Kyuseok Shim. PUBLIC: A decision tree classifier that integrates building and pruning. *Data Mining and Knowledge Discovery*, 4(4):315–344, 2000.

[14] Paul Resnick and Jim Miller. PICS: Internet access controls without censorship. *Communications of the ACM*, 39(10):87–93, 1996.

[15] Cyber Sitter. *At http://www.cybersitter.com/*, online.

[16] V. Vapnic. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[17] James Ze Wang, Jia Li, and Gio Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.

[18] Erik D. Wiener, Jan O. Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval.*

[19] Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49.

[20] Yiming Yang, Seán Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.