Feasibility of Using Machine Learning to Access Control in Squid Proxy Server

Kanchana Ihalagedara, Rajitha Kithuldeniya, Supun Weerasekara and Sampath Deegalla
Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Peradeniya 20400 Sri Lanka
ikihalagedara@gmail.com, rajitha.kithuldeniya@gmail.com, supuniresh@gmail.com and dsdeegalla@pdn.ac.lk

Abstract—**Fast Internet connectivity and billions of websites have made World Wide Web an attractive place for people to use the Internet in their day-to-day life. Educational institutes provide the Internet access to students mainly for educational purposes. However, most of the time, students are allowed to access any content on the web. Therefore, the full bandwidth is consumed due to access to non-educational content such as streaming non-educational videos and downloading large image files, etc.**

Prevention of Internet usage on non-education content is practically difficult due to various reasons. Usually, this is implemented in the proxy server through maintaining a blacklist of URLs. Most of the time, this is a static list of URLs. With the fast growing content on the World Wide Web maintaining a static blacklist is impractical.

In this paper, we propose a methodology to generate dynamic blacklist of URLs using machine learning techniques. We experimentally investigate several machine learning algorithms to predict whether the URL in concern is educational or non-educational. The results of the initial experiments show that linear support vector machines can be used to predict the content with 98.9% accuracy.

A number of users who use the Internet and the number of websites available on the Internet grows rapidly. The rising number of users and the content on the Internet makes managing the Internet in an organization a complex task for administrative users of the network. Managing the Internet is a common problem for companies [14], schools and universities [1]. Therefore, misuse of Internet is a common social dilemma for every society. It has been found that misuse of Internet may affect individual user performance negatively [13].

One of the common misuses of the Internet in educational institutes is accessing non-educational content such as pornography, online gaming, shopping, etc. In an initial investigation in the University of Peradeniya, we have observed more non-educational content access requests than educational in proxy server logs. This observation may indicate that majority of the bandwidth is not consumed for the intended purpose, i.e., learning.

Although there are products available to prevent these misuses, they are unable to cope with the exponential growth of the World Wide Web. Squid proxy server [7] is one such product which caches frequent usage data and can be used to access control. Most of the time, access control is done through a blacklist of URLs, which is a static methodology. Manually updating the list of URLs is time-consuming and error-prone. There are some other products such as SquidGuard [5] which uses external databases to update the blacklist of URLs. However, frequency of updating the databases is a problem. Therefore, in this paper, we are proposing machine learning based dynamically generated blacklisted URLs for non-educational content.

The primary purpose of Squid is to reduce network traffic using caching functionality. However, it also has access controller functionality. Squid ACLs [8] has access classes which can be used to refer a user or groups of users. Squid access operators are used to access control on above classes. ACLs can be used to deny access to a set of URLs and regular expressions can also be used to deny access to URLs based on words. However, having a long list of words will cause performance issues in Squid proxy server, and it does not consider the content of the websites.

SquidGuard [5] is an access controller plugin for Squid. It was developed by Paul Baltzersen and Lars Erik Holland in 1998. ACLs used large lists for creating access controller lists whereas SquidGuard used external databases to keep these lists which make it more efficient than having long lists of URLs. Although SquidGuard is better than Squid ACLs, It does not consider the content of the websites. At present SquidGuards databases are not frequently updated and this makes the blacklists static.

Systems that consider web content when access controlling are called content filters. Dansguardian [6] is an open source web content filter, which is more flexible than SquidGuard. It was created by SmoothWall Ltd. DansGuardian works as a separate unit from Squid and the web browser. It does not rely purely on banned lists of sites. It uses other techniques such as phrase matching, PICS filtering, file extension and file type filtering, etc. After intercepting users request, Dansguardian will check the URL with several lists. If the URL is allowed, Dansguardian will test the web content of the URL. First it will filter the header of the response. Then it will filter HTML text. However, Dansguardian does not dynamically block or filter content. To dynamically block, blacklists or phrase lists should frequently be updated.

Both Dansguardian and SquidGuard do not dynamically block or filter content. They depend on their blacklist or set of words to block or filter content. When filtering, these products does not consider the whole content of the web site, they just filter depending on some words in the website. To dynamically block, their blacklists should be frequently updated.

There were some scholarly works on automated web classification [4], [3]. Patil and Pawar [4] used Naïve Bayes algorithm for automated website classification. Their aim was to automatically categorize organizational web pages into several industrial categories such as academic institutions, hotels, booksellers, etc. They have used several stemming techniques to improve the accuracy of the model. They have also compare the performance of the model by increasing the training data and observed that it increased the accuracy of the final model. Finally, they achieved an accuracy of 89% for all the categories.

Sun et. al. [3] have considered support vector machine algorithm for automated websites classification. In their work, they have also considered context features like title, anchor words of incoming links. They have shown that context features such as links can be used to improve the performance of the classifier significantly.

We are proposing a system that uses machine learning techniques to dynamically update the URL blacklist. The aim of the study is to implement a plugin for the Squid proxy server which will copy the request and extract the URL from the request. Then, the URL will be given to a classification model that decides whether the content of the URL is educational or non-educational. If the system identifies the request as non-educational, the URL will be blacklisted by the system.

Our system should not be overhead for the existing system. Therefore, the latency in our solution should be minimized. If a user is connecting to a new URL through the system, it will take some time to make a decision on the URL. Therefore, even if the user is connecting to an educational site, the server response will be delayed. Then, it will not be the very efficient solution. Therefore, the proposed system will not block any new URLls. The system takes a copy of the request from the server log with the response and then it processes in the background. If the system identifies the request as non-educational, then the system will update the blacklist. When a user tries to access the same URL, access will be denied by the system. The frequency of the update could be decided by the administrative user of the system.

Figure 1 shows the overall architecture of the proposed system. All the components on this system will be plugins for the existing Squid proxy server. Copy request component (3) will copy all the requests by the users. Then preprocess request component will extract the requested URLs. Next, URL will be searched in the database to check whether the URL is previously checked. If the URL is not available in the database (7), web content of the URL is copied and preprocessed (5). Then data will be given to the classification data model (8). If the model predicts the URL as non-educational, URL blacklist in Squid proxy should be updated (10) and URL together with the decision will be stored in the database (9).

We have considered supervised learning approach to create the classification data model. In supervised learning, a learning algorithm requires a labeled dataset. Therefore, we need a set of URLs which labeled as educational or non-educational.

1) Collecting Data: To create a reliable model, we need a relatively large labeled dataset. Manually creating a larger dataset is tiresome. Therefore, we need already categorized dataset. Subject-based web directories like Directory Mozilla (DMOZ) [1], Internet Public Library (ipl2) [10], and Yahoo [11] consist of web pages already classified into various categories such as Business, Art, Science, Education, etc. In these web directories, websites are categorized by experienced editors. In this study, we have considered the DMOZ dataset which consists of XML messages with the details of categorized websites around 4.9 million [1]. Our target was to get the URL and its category. To achieve the target, XML is not a suitable type. Therefore, we have converted these XML messages into JSON objects.

Since our focus is to develop access controller in an educational environment, we have collected a list of URLs accessed within the University of Peradeniya from the Squid proxy server log. By comparing the set of URLs taken from the DMOZ dataset and the server log dataset, we have created a new dataset which contains server log URLs and their categories. This dataset had 322631 URLs.

In this new dataset, there are multiple categories per URL such as education, arts, science, medicine and so on. If an URL falls under educational, we can directly categorize it as educational. However, if URL does not fall under educational, we cannot directly categorize it as non-educational. Since there is no category called non-educational in DMOZ, the URL can be either educational or non-educational. Therefore, we have to categorize the non-educational URLs manually. Choosing non-educational websites can be tricky and subjective. We have considered the following categories as non-educational and manually created a list of URLs for non-educational websites. Non-educational categories were Porn, Live scores, Video games, Tv series, Shopping.

2) Cleaning HTML Content: All URLs are useless without their content for the classification purpose. Here we only considered the text inside the body tag. First text inside the body tag were extracted. Then styles, scripts and all other tags in the text were removed. Next all the English words were extracted. Then all tabs, variables were removed. Next all the words were converted to lower case. Finally we removed any URL which contain less than 50 words after following the previous preprocessing steps on its content.

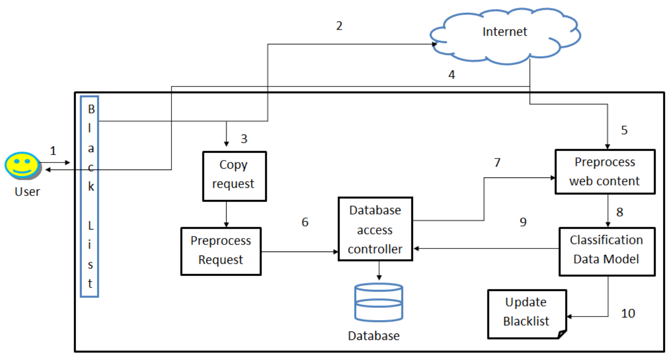

Fig. 1. Architectural design of the proposed system

3) Datasets: We created two datasets from the collected URLs. A small dataset with 200 URLs and a large dataset with 4000 URLs. Each dataset contained the same number of educational and non-educational URLs. We have transformed the content in URLs into bag-of-words representation. We considered word appearance and word frequency for initial experiments with the first dataset. After transforming string into the bag-of-words representation, the first dataset contained 1804 attributes and 200 instances.

4) Generate Data Models: We have used WEKA [12], a data mining tool developed in Java, to generate models from the first dataset. This dataset has URL content and relevant class labels as either educational or non-educational. Based on the algorithms used in the similar context, we have selected Decision Trees (C4.5), Decision Rules (PRISM), Naïve Bayes, and Support Vector Machine to create the data model.

Based on the results of the first dataset, we have considered Naïve bayes and two support vector machine classifiers namely SVM with RBF kernel (SVC) and SVM with linear kernel (Linear SVC) for the second dataset. For this, we have considered Python-based machine learning library Scikit-learn [8] for the implementation.

5) Testing the Data Models: To evaluate the generated data models, we have considered 10-fold cross validation for the first dataset since it contains only 200 URLs. For the larger dataset, we have created a separate independent test dataset with 500 educational URLs and 500 non-education URLs.

A. Principal component analysis (PCA)

When the number of features is high, visualizing data in a hyperplane is not feasible. Therefore, we have transformed the original dimensionality into three principal components.

In Table I list 10-fold cross validation accuracy of the four learning algorithms considered.

TABLE I
10-FOLD CROSS VALIDATION ACCURACY FOR THE FIRST DATASET WITH WORD APPEARANCE

Algorithm	Word Appearance	Word Frequency
C4.5 (J48 in WEKA)	83.0	80.5%
Naive Bayes	95.0	94.5%
PRISM	74.5	63.0%
Support Vector Machines	95.5	94.5%

TABLE II
TEST SET ACCURACY FOR THE SECOND DATASET

Algorithm	Accuracy
Naïve Bayes multinomial	92.9%
SVC	77.5%
Linear SVC	98.9%

Results of the Table I shows that the highest accuracy of the first dataset using 10-fold cross validation is due to Naïve Bayes and SVM classifiers. Decision tree and decision rules give less accuracy than the Naïve Bayes and Support vector machine classifiers. The low accuracy of the decision tree and rules is due to missing some important features when creating both the models.

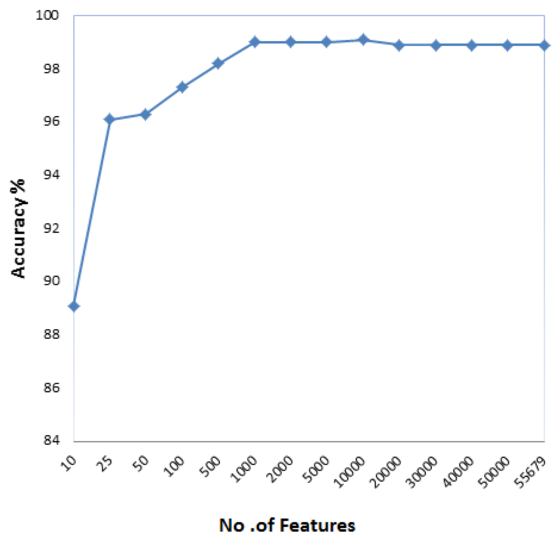

Fig. 2. Accuracy in linearSVC against number of features

Table II shows the results of the second dataset. Here, accuracies are given for the independent test dataset. The results demonstrate that linear SVC gives the best classification accuracy whereas SVC gives much less accuracy. Naïve bayes tends give low accuracy when dataset size gets increased [9].

Figure 2 shows how the accuracy of Linear SVC varies with the number of features considered based on TF-IDF [12], which considers the top maximum features ordered by term frequency across the corpus. We observe an upward trend in accuracy with the number of features considered. However, when we consider 1000 or more features accuracy remained steady. The peak accuracy can be observed with 10000 attributes.

In Figure 3 shows a 3D visualization of the training dataset based on the first three principal components. In blue color, we can see the URLs taken from the educational websites and red color shows the URLs taken from the non-educational websites. There is a clear separation between educational and non-educational URLs. Further, three subgroups of non-educational URLs can be observed. These groupings may be due to non-educational subcategories we have considered during the creation of the training dataset.

In this paper, we have presented a methodology to dynamically prevent non-educational Internet usage for educational institutes. We have developed a machine learning data model to predict whether a given URL is educational or non-educational. We have also presented details of the system architecture to be implemented with Squid proxy server.

The methodology described in this paper can be used to block the non-educational content dynamically. From all the data models, we created and tested LinearSVC yielded the best results. It is scalable and provides quicker results.

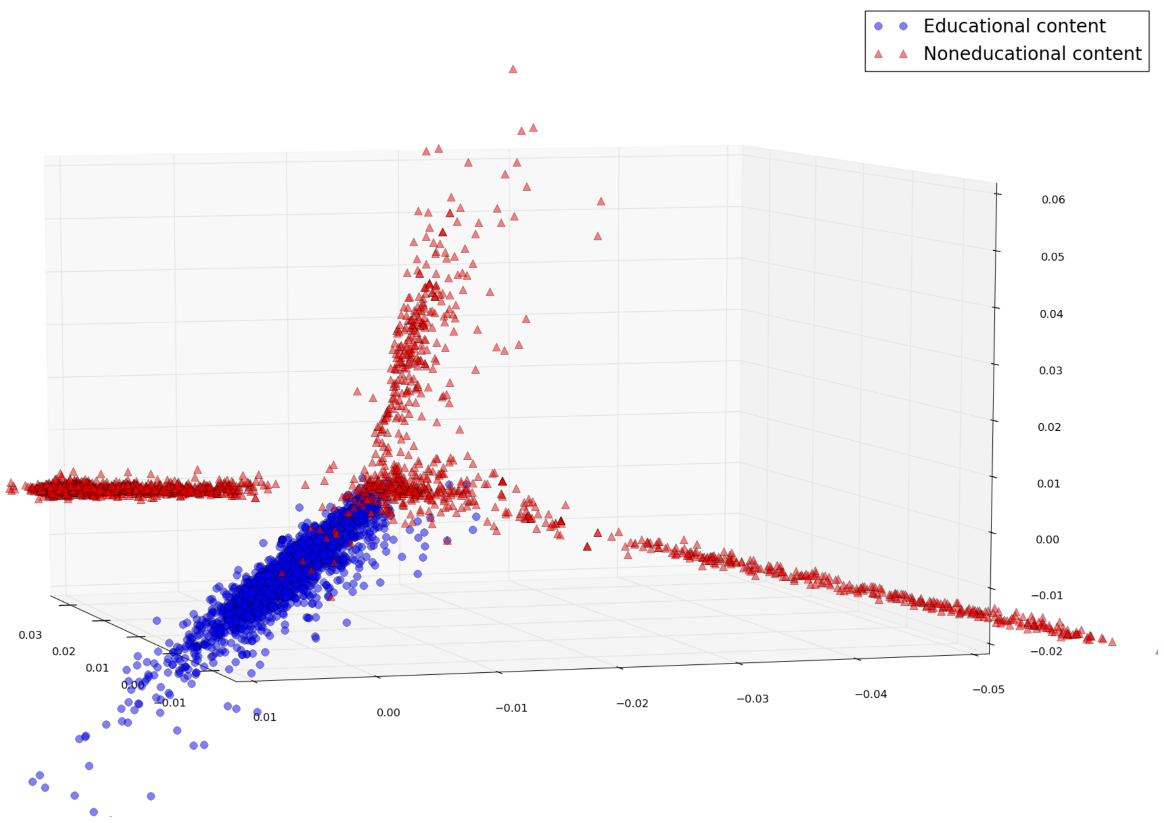

Fig. 3. 3D Visualization of the training data set (4000 urls) by using the first three principal components

As future work, we will implement the proposed system in Squid proxy server as a plugin and test the outcome in a real environment.

We have not considered meta information such as title and other information when blocking non-educational websites. One direction to extend this work is to consider images in websites and meta information for building classification data model.

In this paper, we have only considered content in the English language. Therefore, to extend the study to the local context, we have to collect data from other local languages such as Sinhala and Tamil.

References

[1] DMOZ open directory project. [Online]. Available : http://dmoz.org/

[2] J. M. Pierre, Practical issues for automated categorization of web sites., in *Electronic Proc. of ECDL 2000 workshop on the Semantic Web,* Lisbon, Portugal, 2000.

[3] A. Sun, E. Lim and W. Ng, Web classification using support vector machine, in *Proc. of the 4th Int. workshop on Web information and data management,* McLean, Virginia, USA, 2002, pp. 96 99.

[4] Ajay S. Patil and B.V. Pawar, Automated Classification of Web Sites using Naive Bayesian Algorithm, in *Proc. of the 4th Int. workshop on Web information and data management,* March 14 -16, 2012, Hong Kong.

[5] SquidGuard [Online] Available: http://www.squidguard.org/

[6] DansGuardian [Online] Available: http://dansguardian.org/

[7] Squid [Online] Available: http://www.squid-cache.org/

[8] scikit-learn [Online] Available: http://scikit-learn.org/

[9] Reza Entezari-Maleki, Arash Rezaei, and Behrouz Minaei-Bidgoli, Comparison of Classification Methods Based on the Type of Attributes and Sample Size , *Journal of Convergence Information Technology (JCIT), Vol. 4, No. 3, pp. 94 102,* 2009

[10] Internet public library[Online] Available: http://www.ipl.org//

[11] Yahoo [Online] Available: http://dir.yahoo.com//

[12] I. H. Witten and E. Frank and M. A. Hall. ”Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufmann, San Francisco, 2011.

[13] J. MorahanMartin. ”Internet abuse: Emerging trends and lingering questions.” In A. Barak (Ed.), Psychological aspects of cyberspace: Theory, research, applications (pp. 3269). Cambridge, UK: Cambridge University Press, 2008.

[14] K. S. Young and C. J. Case. ”Employee Internet Management: Current Business Practices and Outcomes”, CyberPsychology and Behavior, 5(4), 355-361, 2002.