SVM Regression for Web Prefetching and Caching

Babita Ujjainiya
Dept of informational technology

Shailendra Kumar Shrivastava
Dept of informational technology

Samrat Ashok Technological Institute Vidisha M.P, India

ABSTRACT
The World Wide Web is a distributed internet system, which provides a many type of services and facilities for users. It increase over the past few year at a very rapidly rate, due to which the amount of traffic over the internet is increasing. As a result, the network performance has now become very slow. The solution of this is to reduce the response time perceived by users. The web prefetching is an effective user prediction technique that extracts useful knowledge from user request sequence. It makes a prediction of the web pages that the user is likely to request in the near future. Web prefetching is one of the effective solutions used to reduce web access latency and improve the quality of service. Many researchers had proposed various predictions based prefetching algorithm and model such as N-gram based prediction [10], PPM model [11], Dynamic prefetching technique [12], and other machine learning technique such as Matrix prefetching [13], Multi dimensional matrix prefetching model [4] and Prediction based model [9]. These techniques are low hit rate and byte hit rate.

The proposed SVM regression used to predict the user's future request for prefetching. This technique to improve system performance and increase hit rate and byte hit rate. Therefore our research is focus to apply SVM technique to the problem of user action for prediction on the web. In particular, we are able to predict the future web page that a user will select. Through the simulation, we found that our approach has quantitative measures such as hit rate and byte hit rate of accessed page.

Keywords
Latency, SVM regression, Prediction, Quantitative, Web Prefetching.

1. INTRODUCTION
The WWW can be considered as a large distributed information system where users can access to shared data objects. Its usage is inexpensive, and accessing information is faster using the WWW than using any other means [1]. The main problem of these may result to extreme congestion on the network and load on the servers, all resulting in unacceptable degradation of the Quality of Service (QOS) at the user end. The web caching is required to alleviate the situation. But the cache management has many challenges in balancing the process of meeting the demands of the users on the one hand and ensuring optimal utilization of system resources on the other hand [9].

Web prefetching is the process of deducing client's future request for web document and getting that document in to the cache, in the background, before an explicit request is made for them. Prefetching capitalizes on the spatial locality present in request streams that is correlated reference for different document and exploits the client's idle time, i.e., the time between successive requests [2]. Web caching is a widely deployed technique in the web architecture that takes advantage of the web object's temporal locality to reduce the user perceived latency. Web caching stores the web objects requested by users. The client side to avoid requesting again the objects to the original web servers [5]. An important advantage of the WWW is that many web servers keep a server access log of its users. These logs can be used to train a prediction model for future document accesses. Based on these models, it can obtain frequent access patterns in web logs and mine association rules for path prediction [3].

Our proposed SVM regression technique predicts a user access request before it is actually demanded. The key issue of SVM regression is to establish an effective user prediction model that extracts useful knowledge from user request sequence and make a prediction of the web pages that the user is likely to request in the near future.

2. BASIC PRINCIPLE
Web caching is a widely deployed technique in the web architecture that takes advantage of the web object's temporal locality to reduce the user perceived latency. Web caching stores the web objects requested by users'. The client side to avoid requesting again the objects to the original web servers [5]. Web prefetching is a technique for reducing web latency based on predicting the next future web objects to be accessed by the user and prefetching them during idle times. This technique takes advantage of the spatial locality shown by the web objects [6]. The prefetching technique has two main components is that the prediction engine and the prefetching engine. The prediction engine runs a prediction algorithm and to predict the next user's request. The prefetching engine handles decide to prefetch [5].

As show in figure 1, the Predictions (PD) are the number of objects predicted by the prediction engine [4]. Prefetch Request (PR) represents the number of objects prefetched. The number of objects prefetched that are requested later by the user is the Prefetch Hit (PH). The opposite of the prefetch hit is the Prefetch Miss (PM), which represents the number of prefetched objects that were never demanded by the user. Finally, User Request (UR) refers to the total amount of objects requested by the user (prefetched or not), and the user request not prefetched represents the number of objects demanded by the user that were not prefetched [4].

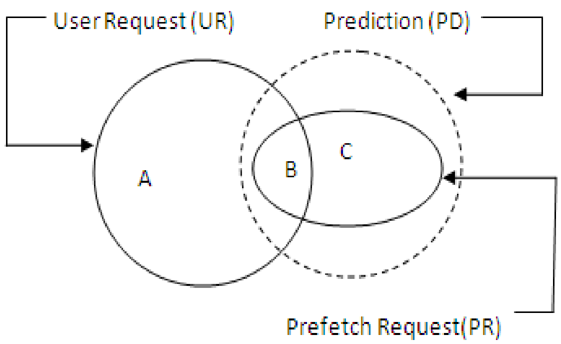


Figure 1: Web prefetching type of requests.

The set of prefetch request is a subset of the prediction set. The result of the intersection between the user request set and prefetch request set is the prefetch hit subset. This subset is the main factor to reduce the perceived latency. In figure.1, A represents a user request not prefetched, which is a user request neither predicted nor prefetched. B is a prefetch request made by the prefetching engine that is requested later by the user, thus becoming a prefetch hit. C is a prefetch miss resulting from an unsuccessful prediction that was prefetched but never demanded by the user. This request becomes extra traffic and extra server load [4].

3. BASIC OF SVM REGRESSION

Support vector machine has been first introduced by Vapnik. The categories for support vector machine is that Support Vector Classification (SVC) and Support Vector Regression (SVR). SVMs are a set of related supervised learning methods originally for pattern recognition and regression. SVM is a learning system using a high dimensional feature space. It prediction functions that are expanded on a subset of support vectors. The main intuition is that given a two-class training set they project its data points in a higher dimensional space and attempt to specify a maximum-margin separating hyperplane between the data points of two classes. This hyperplane is optimal in the sense that it generalizes well to unseen data. A version of a SVM for regression has been proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola. This method is called SVR. The model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close to the model prediction [8].

In the regression [7], estimate the functional dependence of the dependent (output) variable y ϵ R *on* an n-dimensional input variable x. The general regression learning problem is set as follows: the learning machine is given *l* training data from which it attempts to learn the input-output relationship (dependency, mapping or function) f(x). A training data set D={[x(i), y(i)] ϵ R^n*R, i = 1,…..*l*} consists of *l* pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)$, where the inputs x are n-dimensional vectors x ϵ R^n and system responses y ϵ R are continuous values. It introduce all the relevant and necessary concepts of SVM regression in a gentle way starting again with a linear regression hyperplane f(x, w) given as

$$f(x, w) = w^t + b \qquad (1)$$

In eq. (1), f(x,w) is estimation function, w is weight vector, b is threshold. In the case of SVM regression [7], it measures the error of approximation instead of the margin used in classification. The most important difference in respect to classic regression is that use a novel loss (error) functions.

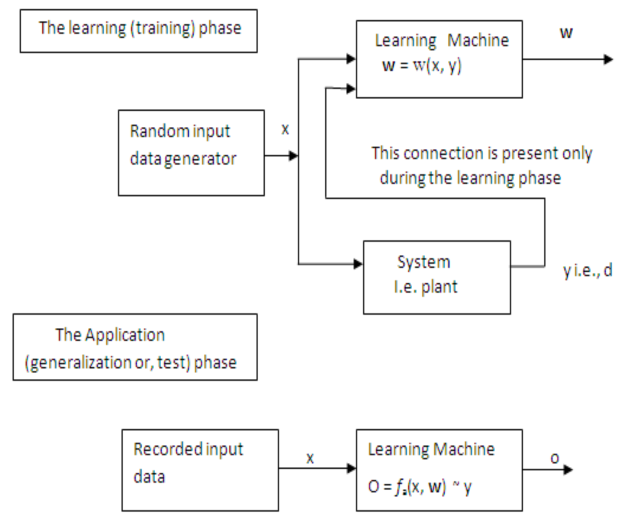


Figure 2: SVM learning and testing phase model.

As in figure 2, the three basic components in all learning from data tasks [7]. First, a generator of random inputs x, second a system whose training responses y(*d*) are used for training the learning machine, and third, a learning machine which, by using inputs x_i and system's responses y_i, should learn the unknown dependency between these two sets of variables defined by the weight vector w. During the training phase a learning machine should be able to find the relationship between an input space X and an output space Y, by using data D in regression tasks [7].

In our technique input as a number of users requests and then predict next future requests, that is actual output.

4. PROPOSED MODEL AND ALGORITHM

Several researchers have previously studied the use of a prediction system for prefetching. For presents a new supervised machine learning algorithm SVM regression used to implement web prefetching algorithm.

In SVM regression prediction model improves the generation and delivery of web content, it is employed to implement an prefetching system. The purpose is also to directly correlate prediction to performance. Each session of the test request has a cache associated with it. SVR prediction model is located nearest of web server system. SVR is predicting next future request. Server has a various web log files such a web page. Firstly, the request is sent by client to the web server and waits for response. At the beginning the cache is empty. For the request of each session, the system first looks for that request is in the cache. SVR prediction is applies with request page on server and then fetch next page. With this request the web server sends the requested web page such as predicted page to the cache. If the prediction is correct, sends a request, the page in the cache and then calculate hit rate and byte hit rate. For next new session of request then the cache will again send the request to the server, which applies the SVR prediction model to send the next future page.

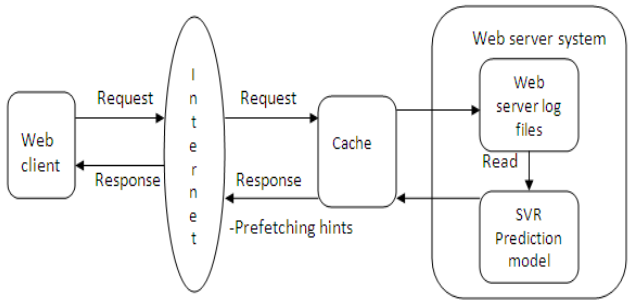


Figure 3: Proposed web prediction model.

Step of proposed algorithm is following:

1. Given current data set d then grouped into two dataset called d1 and d2. Both d1 and d2 data set group are differently used. d1 dataset for training client request as input and d2 dataset for testing.
2. Classified both dataset d1 and d2 into separate filed.
3. Train dataset d1 using support vector train on feature plane.
4. Contract regression hyperplane that read request of client.
5. Apply n client request for prediction. SVR are using for testing that request.
6. Server checking that client request with in d2 Dataset and fetch next request for client.
7. Cache is empty then store predicted requests.
8. Calculate hit rate, byte hit rate, and cache size with predicted requests.
9. Repeat that step3 to step8 with next new requests.

5. EXPERIMENTS AND RESULT ANALYSIS

The proposed system has been implemented and tested in Mat Lab version 7.5 under Microsoft Windows XP operating system. The dataset collect from a NASA kennedy space centers web server in florida. This web log contains all the HTTP requests collected from 00:00:00, 1 July 1995, to 23:59:59, 17 July 1995.This web log contains all the HTTP requests. In our experiment use for total 21000 requests. We have showed that predictive system performance in term of hit rate and byte hit rate. Both hit rate and byte rate are growing in our experiment. In figure 4 show as NASA web logs information, it is sample of our dataset.

uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0

slppp6.intermind.net - - [01/Aug/1995:00:00:10 -0400] "GET /history/skylab/skylab.html HTTP/1.0" 200 1687

piweba4y.prodigy.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/launchmedium.gif HTTP/1.0" 200 11853

slppp6.intermind.net - - [01/Aug/1995:00:00:11 -0400] "GET /history/skylab/skylab-small.gif HTTP/1.0" 200 9202

Figure 4: Web log information of NASA dataset.

All the experiments here have been executed on the NASA Dataset, which is divided into two groups i.e. dataset d1 and dataset d2. Both datasets are then classified into five separate fields. Dataset d1 used for client request as an input for training and dataset d2 used for matching. Each field contains different data. In fig.5 first field represent 'request client ID', second field represent 'time period' i.e. time stamp, third represent 'request list', fourth represent 'reply list' i.e. reply of the sever and last represent 'size of file' in byte. In simulation, each field used as a table form. Data is then converted from string to numeric value. SVM Regression trains data field of d1. After the training process dataset is tested and N no of requests are applied for prediction. SVM prediction processes take that N no of requests and then predict next future request for client. Predicted requests are then stored into buffer as a cache. Predicted requests are then matched with dataset d2 and counted. This process is called 'hit'. Hit rate, byte-hit rate and cache size are then calculated. Again, next N no of new requests are applied for prediction via SVM Prediction, then next future request for client are generated and again Hit rate ,Byte hit rate and Cache size are calculated. SVM prediction are run six time with different set of request and generate predict requests for future.

SVM Regression used to predict the user's next request in web prefetching by extracting useful knowledge from historical user requests. The number of web pages on the Internet has grown explosively in the past and such growth is expected to be more acute in the future. Web prefetching is one of the solutions used to reduce web access latency. SVR is a regression technique based on support vector machine, a very effective and mathematically well founded machine learning approach. SVR is a new generation of Machine Learning algorithms for predictive data modeling problems. As a positive aspect, SVR can be easily integrated with kernel functions allowing the learning mechanism to adapt to different datasets.

5.1 Performance metrics

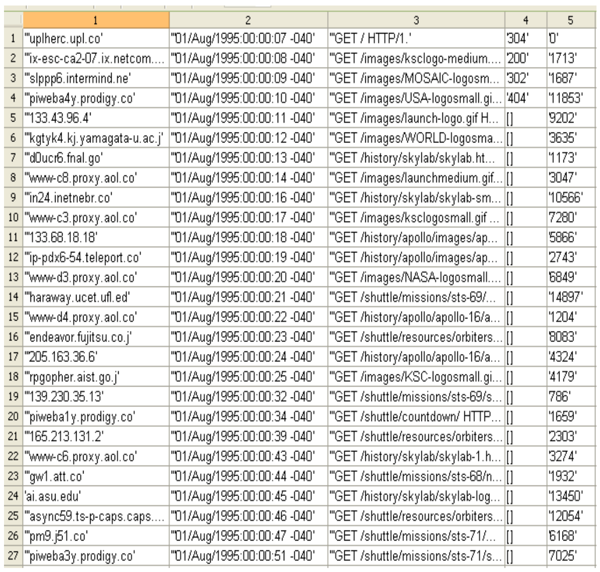
The hit rate is the ratio between the number of requests that hit in the cache and the total number of requests.

The byte-hit rate is an even more realistic measure of performance for web caching, it is the ratio between the number of bytes that hit in the proxy cache and the total number of bytes requested sent to its clients.

The cache size is the ratio of predicted requests from cache and the dataset. The probability of predict client request that can be found in cache as increases cache size. The cache size indicates the amount of space available for storing web objects.

5.2 Testing the System

The following output shows the results of testing when run. In table requests show number of requests of dataset are apply for prediction. We examine each set of requests for 6 times that run on SVM prediction. Completion of prediction process then calculates hit rate, cache size and byte hit rate. These calculations are based on predicted requests. When SVM prediction predict more request and then increase hit rate and byte rate.



	1	2	3	4	5
1	'"uplherc.upl.co'	'"01/Aug/1995:00:00:07 -040'	'"GET / HTTP/1.'	'304'	'0'
2	'"ix-esc-ca2-07.ix.netcom....	'"01/Aug/1995:00:00:08 -040'	'"GET /images/ksclogo-medium....	'200'	'1713'
3	'"slppp6.intermind.ne'	'"01/Aug/1995:00:00:09 -040'	'"GET /images/MOSAIC-logosm...	'302'	'1687'
4	'"piweba4y.prodigy.co'	'"01/Aug/1995:00:00:10 -040'	'"GET /images/USA-logosmall.gi...	'404'	'11853'
5	'"133.43.96.4'	'"01/Aug/1995:00:00:11 -040'	'"GET /images/launch-logo.gif H...	[]	'9202'
6	'"kgtyk4.kj.yamagata-u.ac.j'	'"01/Aug/1995:00:00:12 -040'	'"GET /images/WORLD-logosma...	[]	'3635'
7	'"d0ucr6.fnal.go'	'"01/Aug/1995:00:00:13 -040'	'"GET /history/skylab/skylab.ht...	[]	'1173'
8	'"www-c8.proxy.aol.co'	'"01/Aug/1995:00:00:14 -040'	'"GET /images/launchmedium.gif...	[]	'3047'
9	'"in24.inetnebr.co'	'"01/Aug/1995:00:00:16 -040'	'"GET /history/skylab/skylab-sm...	[]	'10566'
10	'"www-c3.proxy.aol.co'	'"01/Aug/1995:00:00:17 -040'	'"GET /images/ksclogosmall.gif ...	[]	'7280'
11	'"133.68.18.18'	'"01/Aug/1995:00:00:18 -040'	'"GET /history/apollo/images/ap...	[]	'5866'
12	'"ip-pdx6-54.teleport.co'	'"01/Aug/1995:00:00:19 -040'	'"GET /history/apollo/images/ap...	[]	'2743'
13	'"www-d3.proxy.aol.co'	'"01/Aug/1995:00:00:20 -040'	'"GET /images/NASA-logosmall....	[]	'6849'
14	'"haraway.ucet.ufl.ed'	'"01/Aug/1995:00:00:21 -040'	'"GET /shuttle/missions/sts-69/...	[]	'14897'
15	'"www-d4.proxy.aol.co'	'"01/Aug/1995:00:00:22 -040'	'"GET /history/apollo/apollo-16/a...	[]	'1204'
16	'"endeavor.fujitsu.co.j'	'"01/Aug/1995:00:00:23 -040'	'"GET /shuttle/resources/orbiters...	[]	'8083'
17	'"205.163.36.6'	'"01/Aug/1995:00:00:24 -040'	'"GET /history/apollo/apollo-16/a...	[]	'4324'
18	'"rpgopher.aist.go.j'	'"01/Aug/1995:00:00:25 -040'	'"GET /images/KSC-logosmall.gi...	[]	'4179'
19	'"139.230.35.13'	'"01/Aug/1995:00:00:32 -040'	'"GET /shuttle/missions/sts-69/s...	[]	'786'
20	'"piweba1y.prodigy.co'	'"01/Aug/1995:00:00:34 -040'	'"GET /shuttle/countdown/ HTTP...	[]	'1659'
21	'"165.213.131.2'	'"01/Aug/1995:00:00:39 -040'	'"GET /shuttle/resources/orbiters...	[]	'2303'
22	'"www-c6.proxy.aol.co'	'"01/Aug/1995:00:00:43 -040'	'"GET /history/skylab/skylab-1.h...	[]	'3274'
23	'"gw1.att.co'	'"01/Aug/1995:00:00:44 -040'	'"GET /shuttle/missions/sts-68/n...	[]	'1932'
24	'ai.asu.edu'	'"01/Aug/1995:00:00:45 -040'	'"GET /history/skylab/skylab-log...	[]	'13450'
25	'"async59.ts-p-caps.caps....	'"01/Aug/1995:00:00:46 -040'	'"GET /shuttle/resources/orbiters...	[]	'12054'
26	'"pm9.j51.co'	'"01/Aug/1995:00:00:47 -040'	'"GET /shuttle/missions/sts-71/...	[]	'6168'
27	'"piweba3y.prodigy.co'	'"01/Aug/1995:00:00:51 -040'	'"GET /shuttle/missions/sts-71/s...	[]	'7025'

Figure 5: Partial information of NASA web log.

Table 1. Performance Analysis using SVM regression

Requests	Hit Rate	Cache Size	Byte Hit Rate
1000	66.6481	0.0136	66.5100
2000	74.5864	0.0239	88.8722
4000	81.1205	0.0459	93.0043
7000	84.9020	0.0760	95.7791
80000	85.7083	0.0870	96.1029
130000	88.5166	0.1412	98.1029

5.3 Performance Graph of SVM Regression

The following graphs show the result of predicted request in performance graph. As in see figure 6 show hit rate and cache size of SVM regression. The hit rate is 67 up to 89 % and cache size is 0.01 up to 0.14. In figure 7 show byte hit rate is 67 up to 98 %. Both graphs are based on SVM regression. There are six iterations of different request of result according to hit rate and cache size.

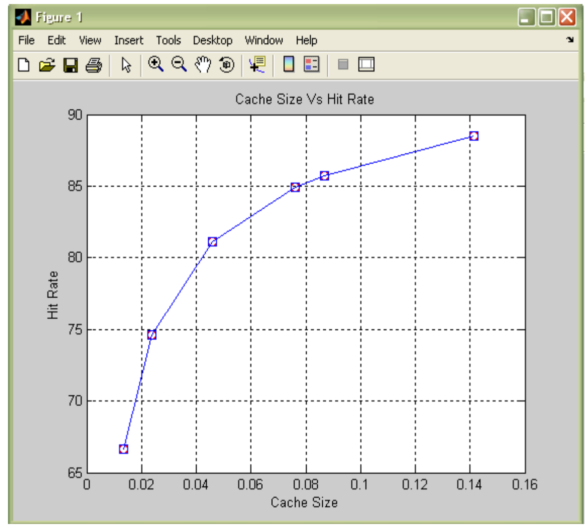


Figure 6: Hit rate and cache size of SVM regression.

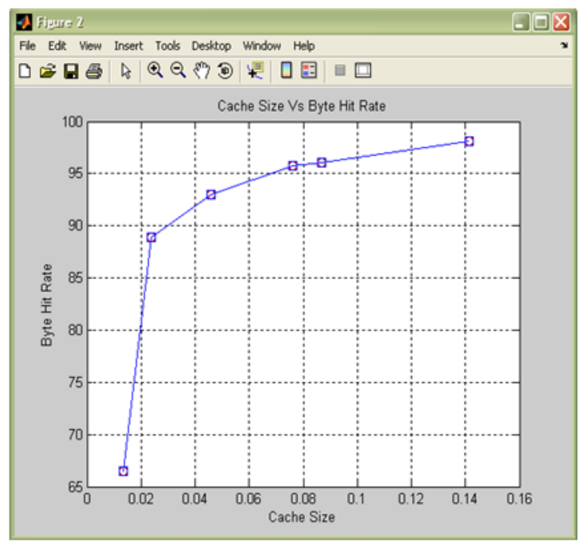


Figure 7: Byte hit rate and cache size of SVM regression.

6. CONCLUSION AND FUTURE WORK

The World Wide Web is large distributed information system where users can access shared data object. Increasing popularity of the World Wide Web over the past few years has imposed a significant traffic burden upon the internet. AS the result, the internet services are slow down such as retrieve page from server and other decrease the performance of system. To solve this problem, we have applied supervised machine learning technique SVM regression for prediction in web prefetching. The whole procedure had examined with SVM regression for prediction of requests and calculate hit rate and byte hit rate for each iterations. The hit rate is varied from 67% to 89% and byte hit rate is varied from 67% to 98%. The performance found is efficient. Using this technique improve system performance in term of hit rate and byte rate.

In future there can be some other ways to improve the performance of the system and training time, one such as Incremental Support Vector Machine (ISVM).

7. REFERENCES

[1] Sarina sulaiman siti, Ajith Abraham shahida Sulaiman. 2008. Web caching and prefetching what, why, and how. IEEE.

[2] C. Umapathi and J. raj. 2006. Prefetching algorithm for improving web cache performance. Journal of application science asian network scientific information, pp. 3122-4127.

[3] Q. Yang, H. H. Zhang and T. Li. 2001. Mining web logs for prediction models in www caching and prefetching. 7th ACM International Conference, pp. 473-478.

[4] Wenying Feng, Karan Vij .2008. Web cache prefetching by multi - dimensional matrix. IEEE, pp. 265-270.

[5] Johann Marquez, Josep Domenech, Jose A. Gil and Ana pont. 2008. A web caching and prefetching simulator. IEEE, pp. 346- 350.

[6] Johann Marquez, Josep Domenech, Jose A. Gil and Ana pont. 2008. An intelligent technique for Controlling Web prefetching costs at the server side. IEEE/ACM, pp. 669-975.

[7] Janusz Kacprzyk. Support Vector Machine: Thory and Aplications. Studies in Fuzziness and Soft Computing, Volume 177. pp 35-38.

[8] Debasish Basak, Srimanta Pal and Dipak Chandra Patranabis. 2007. Support Vector Regression. IEEE, vol. 11, October, pp. 203-224.

[9] Areerat songwattana. 2008. Mining web logs for prediction in prefetching and caching. IEEE, pp. 106-1011.

[10] Qiang Yang and Haining Henry Zhang. 2003. web-log mining for predictive web caching. IEEE, pp. 1050-1053.

[11] Zhijie ban, Zhimin GU, Yu Jin. 2008. A PPM prediction model based on stochastic gradient descent for web prefetching. IEEE, pp. 166-173.

[12] Achuthsankar S. Nair, Jayasudha J.S. 2007. Dynamic web prefetching technique for latency reduction. IEEE, pp. 202-206.

[13] Wenying Feng and Karan Vij. 2007. Machine learning Prediction and web access modeling. IEEE, pp. 607-612.