Middle-East Journal of Scientific Research 24 (6): 2099-2109, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.06.23563

Significance of Combining Machine Learning Techniques for Web Caching with Web Pre-Fetching

[1]*V. Vijilesh*, [2]*R. Nedunchezhian and* [3]*K.R. Baskaran*

[1]Department of IT, Kumaraguru College of Technology, India
[2]Department of CSE, Kalaignar Karunanidhi Institute of Technology, India
[3]Department of IT, Kumaraguru College of Technology, India

Abstract: The performance of web-based system in terms of access delay is improved by deploying the web caching strategy. Web caching is the technique in which frequently used web objects can be kept closer to the web user. Apart from experiencing lesser access delay, caching process also helps in better bandwidth utilization, reduction of network congestion and to reduce load on the origin server. Web pre-fetching is the process of bringing the web objects from the origin server which has more likelihood of being used in the future. Web pre-fetching helps in increasing the cache hits and reducing the user perceived latency. Combining web caching and pre-fetching techniques results in experiencing further less access delay and better bandwidth utilization than using the above two techniques individually. In this work, intelligent web caching techniques namely, SVM–LRU, naive Bayesian LRU and Neuro-Fuzzy LRU are used in combination with clustering technique for web pre-fetching. Comparison of the above intelligent caching techniques combined with pre-fetching technique is done using the metrics Hit Ratio (HR) and Byte Hit Ratio (BHR). For clustering, both inter-clustering and intra-clustering are considered. It has been demonstrated that combining SVM-LRU caching technique with intra clustering technique results in higher HR and BHR than the other combinations for both Inter and Intra-clustering.

Key words: Machine Learning · Clustering · Hit Ratio · Byte Hit Ratio · LRU · Access Latency

INTRODUCTION

The volume of data available on the Internet has grown rapidly due to exponential growth of web users. In spite of advances in technology, huge traffic is leading to access delays [1]. Web caching and web pre-fetching are two techniques that can be used to reduce the access delays. Web caching is the process of keeping web objects in near-by user locality, which can be accessed quickly. Web pre-fetching is the process of bringing the web objects from the origin server into the web cache before a request is made for it. Combining Web caching and Web pre-fetching techniques further reduces the access latency.

Related Work: Web caching and Web pre-fetching: Web caching is a widely deployed technique in taking advantage of the web object's temporal locality for reducing the access latency. The popular performance metrics used in web caching are HR and BHR. A cache-hit ratio is the number of searched data found in the cache divided by the total number searched data. The higher the hit ratio, the better is the performance. Traditionally cache replacement techniques like Least Recently Used (LRU), Least Frequently Used (LFU) are used for web object replacements if sufficient space is not found in the cache for storing a new web object. These conventional page replacement techniques lead to cache pollution problem and hence it is wise to use intelligent web caching techniques to improve HR and BHR. In general, cache replacement policies are classified as regency-based, frequency-based, size-based, function-based and randomized replacement policies.

Web pre-fetching is a popularly used technique to capitalize spatial locality of the web objects. Web pre-fetching is the process of anticipating future page requests of users and loading them into the cache in advance. Combining web caching and web pre-fetching

Corresponding Author: V. Vijilesh, Department of IT, Kumaraguru College of Technology, India.

techniques will result in increased HR and reduced user-perceived latency. However, if the above two techniques are integrated inefficiently it would lead to increase in network traffic and overloads the web server. Prediction algorithms used for improving the precision of pre-fetched objects are classified based on object popularity [2], Markov models [3-5], web structure [6, 7] and predication by partial matching [8-10], data mining [11] and genetic algorithms [12].

Intelligent Machine Learning Techniques: The availability of web access log files that can be used as training data is the main idea behind using intelligent web caching approaches. The intelligent web caching techniques that are found in the literature is described below:

Datta *et al*. [1] have acknowledged web caching to be a prominent research area because of its usage in reducing access delays while accessing the internet. Mookherjee and Tan [13] have proposed an analytical framework for using LRU cache replacement technique and their framework evaluates the latter replacement method under different cache size requirements. T Koskela *et al.* [14] proposed a nonlinear model for optimizing web cache performance that predicted values for each cache object by using features such as HTTP responses of the server and access log file. Jake Cobb and Hala ElAarag [15, 16] experimented the usage of back-propagation neural networks for cache replacement. Calzarossa [17] proposed the usage of fuzzy logic for cache replacement. Farhan [18] has proposed BPNN (back-propagation neural network) for making caching decision and using LRU replacement policy.

In the model based predictive pre-fetching method proposed by Yang *et al.* [19], web caching and web pre-fetching techniques were integrated. Craswell *et al.* [20], brings out the effectiveness of content-based ranking method in finding the web sites that will be accessed by the web user in future is analyzed. In that experiment, importance of anchor texts in finding useful web sites is reported. Keyword-based semantic pre-fetching technique is proposed by Cheng-Zhong Xu and Tamer I. Ibrahim [21] where neural networks are used for future prediction of user requests based on past web objects retrieval. Alexander P. Pons [22], proposed semantic link pre-fetcher that makes use of semantic link information linked with the current web page hyper links for future prediction of useful web pages. The significance of textual information in the visited pages and in the followed links for determining the user's preference is proposed by Georgakis and Li [23].

Proposed Methodology: User log files are used for identifying the web objects accessed by various users in a given time interval. After preprocessing the log file contents, classification of the web objects into Class 0 and Class 1 are done using the metrics: recency, frequency, retrieval time and web object size. SVM classifier is trained using the above features. WNG is constructed for every thirty minutes for each user, based on the user's access patterns. Choosing lesser time interval for WNG formation results in overloading of the server and longer time interval results in longer time duration for pre-fetching popular objects. Website URLs are represented by nodes in the WNG and the edges in the WNG is used to represent the transitions made by a user between two URLs. An inter-site clustering represents the clustering of home pages of various URLs browsed by a user in the specified time interval. An intra-site clustering includes clustering of all referred pages belonging to every web site in the specified time interval. Confidence and Support parameters are used to identify the frequently browsed web objects of a user. A threshold value is fixed for the above parameters. The edges that have lesser values than the threshold value are removed. BFS (Breadth First Search) algorithm is then applied for cluster formation.

Cache memory is partitioned into two namely short-term cache and long-term cache. Pre-fetched web objects are brought into short term cache. Whenever the user requested web object is found in the short term cache, its count of access is increased by one. If sufficient space is not available in the short term cache, LRU page replacement technique is deployed to remove pages for space creation. When the access count of a web object becomes greater than the threshold value, it is classified using SVM/Neuro-fuzzy/Bayesian classifier. Web objects classified as Class 0 are moved to the bottom of long term cache and objects classified as Class 1 are moved to the top of long term cache.

Baskaran *et al.* [24, 25] have proposed to remove web objects from long term cache when sufficient space is not available in long term cache for moving a web object from short term cache.

In order to improve the efficiency of the above technique, web objects identified for replacement from long term cache are moved to short term cache. The reason is that, objects stored in long term cache are frequently accessed web objects. Hence their removal may result in cache miss in the immediate future. To avoid this short coming, web objects from long term cache are moved to short term cache. If sufficient space is not available in short term cache, LRU replacement technique is used to remove web objects to create necessary space in the short term cache for bringing the web object from long term cache. The dataset chosen for experimenting the above process is taken from the URL ftp://ftp.ircache.net/Traces/DITL- 2007-01-09/. The dataset in the above URL contains the details and properties of web objects browsed by number of users. The proposed intelligent Web caching algorithm is shown below:

Step 1: The user requests for the web object

Step 2: Check whether the web object is available in the short term cache, if so go to Step 3, otherwise go to Step 4.

Step 3: Read the web object available in the short term cache, increase the access count by 1 and go to Step 8.

Step 4: Check whether the web object is available in the long term cache, if so go to Step 5 otherwise go to Step 6.

Step 5: Read the web object available in the long term cache and go to Step 8.

Step 6: Read the web object available in the origin server.

Step 7: Add the web object into the short term cache.

Step 8: Send the web object to the processor.

Step 9: Check whether the access count of a web object in short term cache is greater than the threshold. If so, go to Step 10 otherwise go to Step 14.

Step 10: Classify the web object using SVM/Bayesian/ Neuro-fuzzy classifier.

Step 11: Check whether the classifier returns 0, if so go to step 12 otherwise go to Step 13.

Step 12: The web object is classified as Class 0 and is moved to bottom of the long term cache and go to Step 14.

Step 13: The web object is classified as Class 1 and is moved to the top of long term cache.

Step 14: Check for space availability in the long term cache, if space not available, go to Step 15 otherwise go to Step 18.

Step 15: Remove web objects from the lower part of the long term cache until sufficient space is created.

Step 16: Add the removed web objects from long term cache into short term cache.

Step 17: Check for space availability in the short term cache, if space is not available, objects are removed by using LRU replacement technique.

Step 18: End of the process.

Experimental Results: HR and BHR are computed for various combinations of support and confidence values. The results are plotted in the graphs shown below in Figures 1(a) to 4(d):

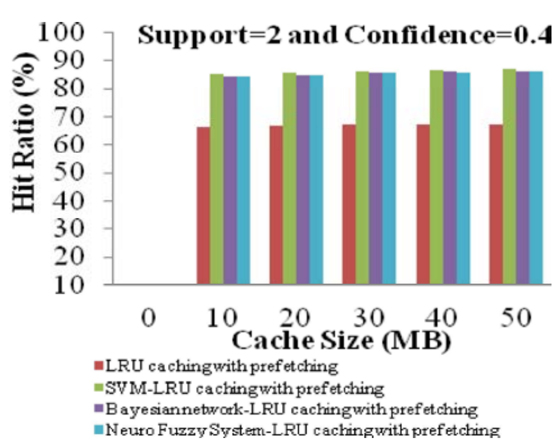

Fig. 1(a): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=2 and Confidence= 0.4

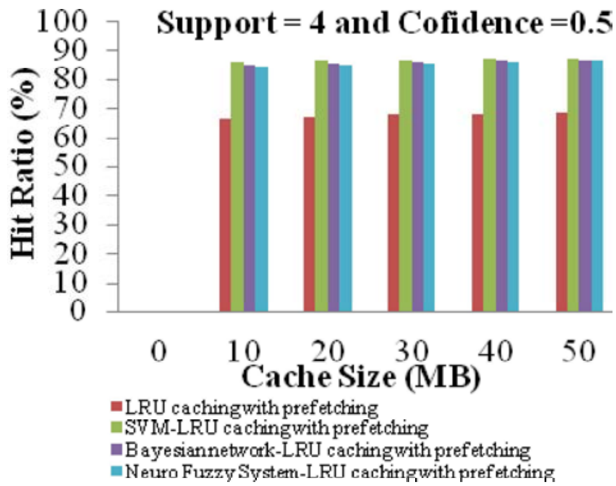

Fig. 1(b): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=4 and Confidence= 0.5

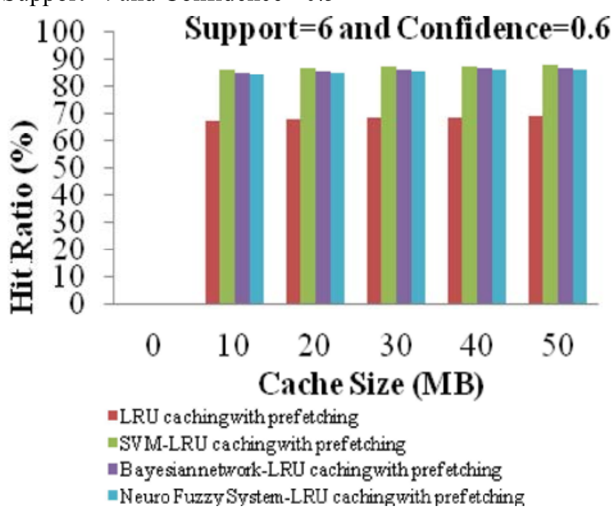

Fig. 1(c): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=6 and Confidence= 0.6

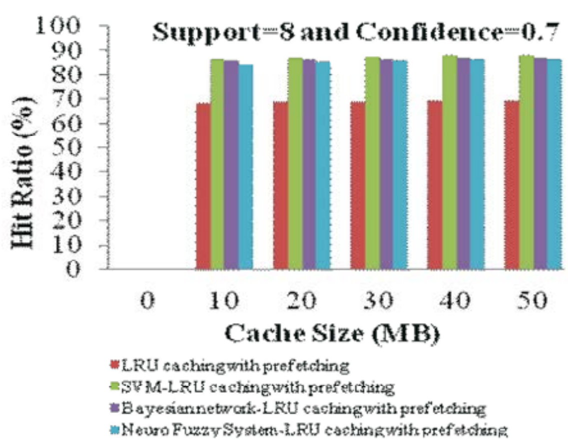

Fig. 1(d): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=8 and Confidence= 0.7

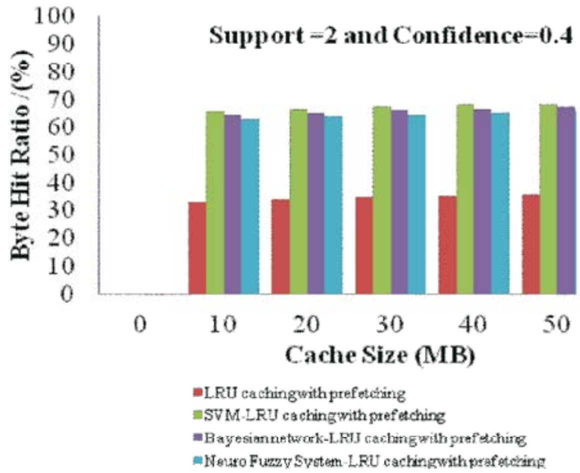

Fig. 2(a): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=2 and Confidence= 0.4

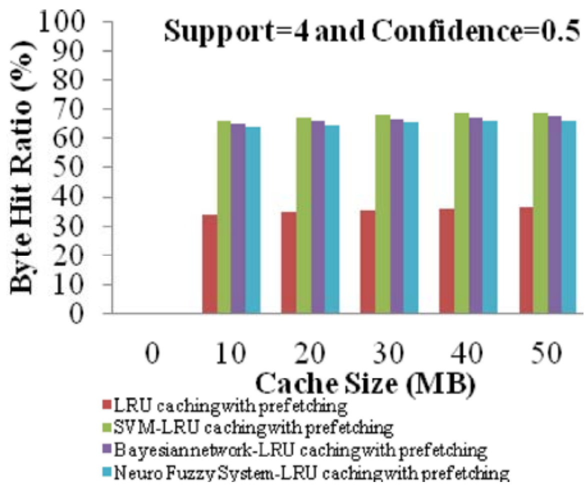

Fig. 2(b): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=4 and Confidence= 0.5

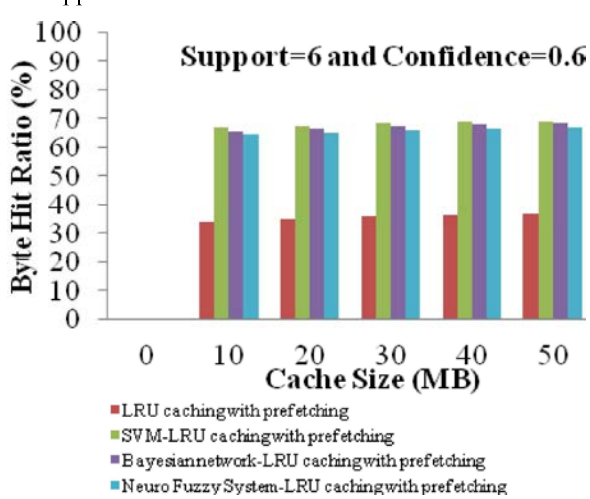

Fig. 2(c): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=6 and Confidence= 0.6

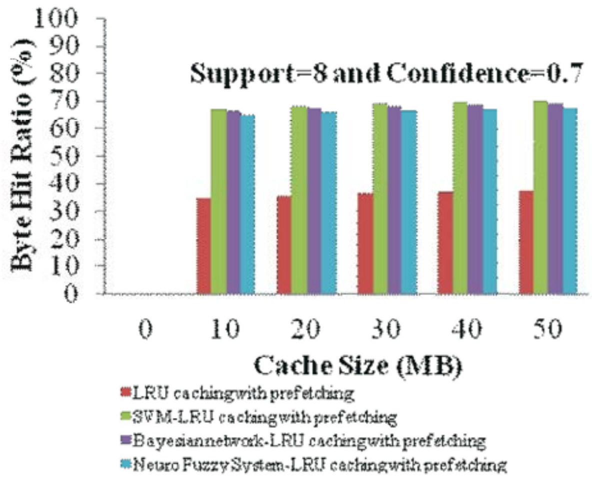

Fig. 2(d): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (inter-clustering) for Support=8 and Confidence= 0.7

The average values of HR and BHR achieved for various combined caching and pre-fetching techniques (inter-clustering) by choosing different set of values for Support and Confidence is summarized in Table 1:

Table 1: Comparison of HR and BHR achieved for various values for Support and Confidence parameters using combined Intelligent web caching and Web pre-fetching (inter-clustering)

	Support (S) and Confidence values (C)							
	S=2, C=0.4		S=4, C=0.5		S=6, C=0.6		S=8, C=0.7	
Machine Learning Techniques	HR (%)	BHR (%)	HR (%)	BHR (%)	HR (%)	BHR (%)	HR (%)	BHR (%)
LRU	67.0	34.5	67.6	35.2	68.4	35.8	69.0	36.1
Neuro-fuzzy-LRU	85.2	64.4	85.4	65.3	85.6	66.0	85.8	66.3
Bayesian network-LRU	85.4	65.9	85.8	66.7	86.2	67.4	86.6	67.9
SVM-LRU	86.1	67.2	86.6	67.9	87.1	68.5	87.4	68.7

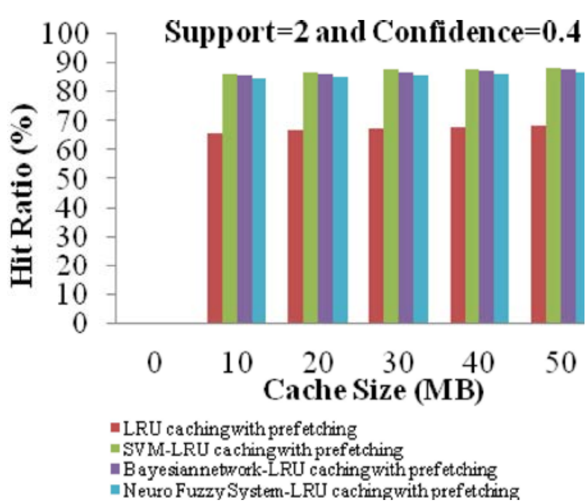

Fig. 3(a): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=2 and Confidence= 0.4

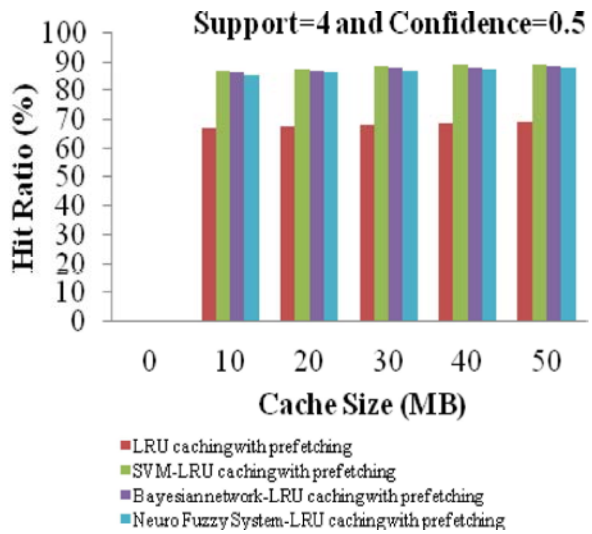

Fig. 3(b): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=4 and Confidence= 0.5

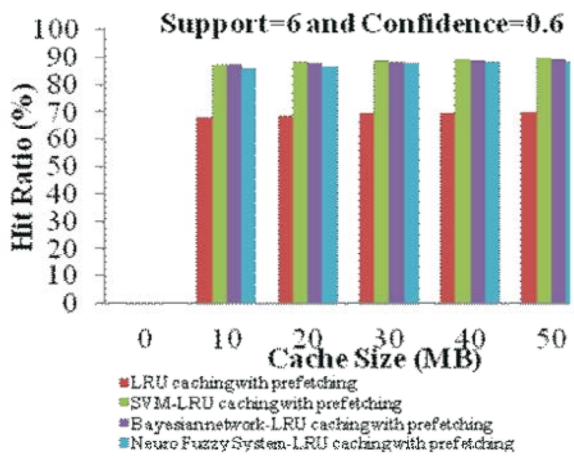

Fig. 3(c): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=6 and Confidence= 0.6

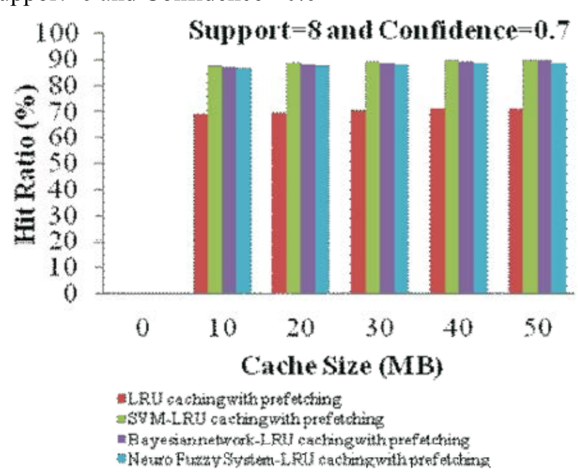

Fig. 3(d): Analysis of HR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=8 and Confidence= 0.7

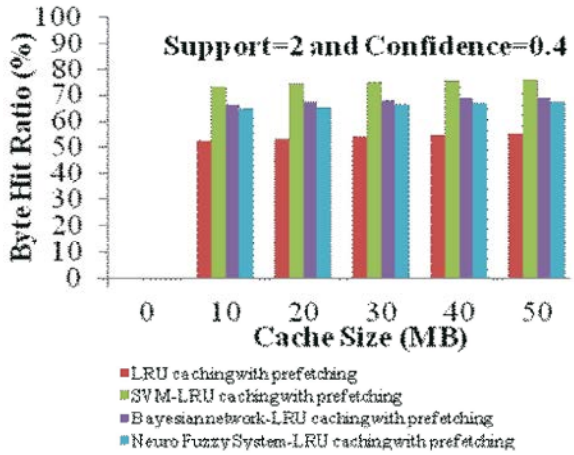

Fig. 4(a): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=2 and Confidence= 0.4

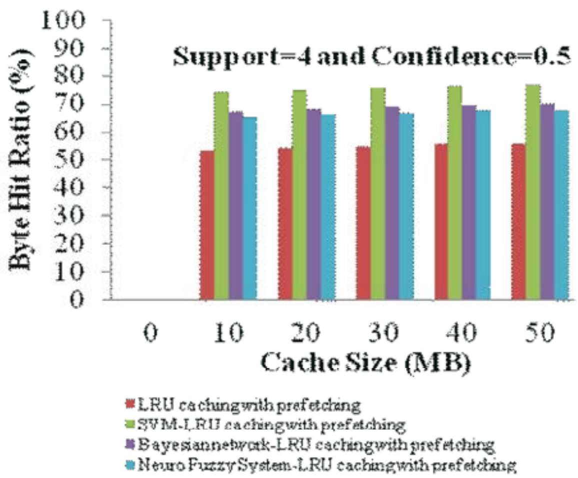

Fig. 4(b): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=4 and Confidence= 0.5

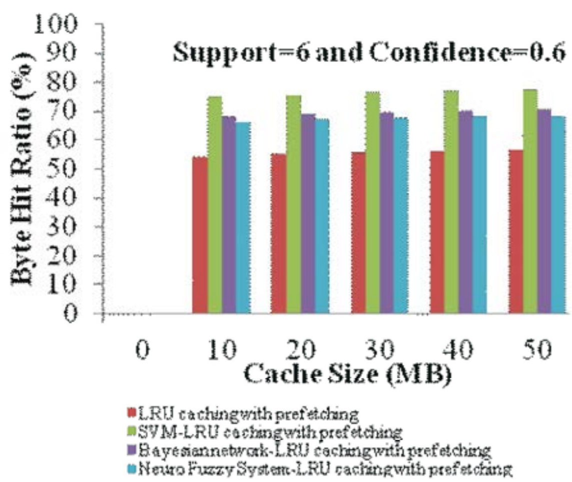

Fig. 4(c): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=6 and Confidence= 0.6

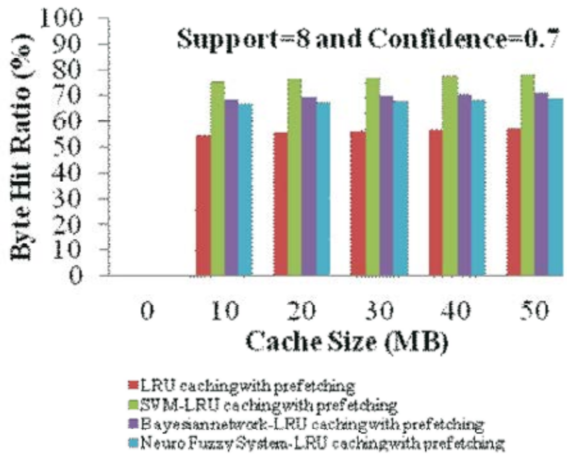

Fig. 4(d): Analysis of BHR considering various intelligent web caching techniques combined with web pre-fetching (intra-clustering) for Support=8 and Confidence= 0.7

The average values of HR and BHR achieved for various combined caching and pre-fetching techniques (intra-clustering) by choosing different set of values for Support and Confidence is summarized in Table 2:

Table 2: Comparison of HR and BHR achieved for various values for Support and Confidence parameters using combined Intelligent web caching and Web pre-fetching (intra-clustering)

	Support (S) and Confidence values (C)							
	S=2,	C=0.4	S=4,	C=0.5	S=6,	C=0.6	S=8,	C=0.7
Machine Learning Techniques	HR (%)	BHR (%)	HR (%)	BHR (%)	HR (%)	BHR (%)	HR (%)	BHR (%)
LRU	67.4	53.8	68.2	54.7	69.2	55.4	70.2	56.1
Neuro-fuzzy-LRU	85.8	66.0	86.7	66.7	87.4	67.4	88.1	67.9
Bayesian network-LRU	86.7	67.9	87.6	68.7	88.4	69.4	88.7	70.0
SVM-LRU	87.4	74.7	88.1	75.5	88.7	76.2	89.1	76.8

From the experiments conducted it is observed that there is improvement in HR and BHR achieved when intra-clustering is replaced with inter-clustering technique. It is also demonstrated that SVM-LRU caching with pre-fetching technique results in higher HR and BHR when compared with other intelligent web caching techniques.

The average increase in HR and BHR for various caching techniques combined with clustering technique for pre-fetching is tabulated below:

Table 3: Average increase in HR and BHR achieved by using intra-clustering over inter-clustering

% improvement achieved in using intra-clustering				
	LRU caching with pre-fetching	Neuro-fuzzy-LRU caching with pre-fetching	Bayesian-LRU caching with pre-fetching	SVM-LRU caching with pre-fetching
HR	0.75	1.40	1.80	1.50
BHR	19.6	1.50	2.00	7.60

From Table 3, it is demonstrated that, while using intelligent web caching techniques, SVM-LRU caching with pre-fetching results in substantial increase in BHR.

CONCLUSION

From the above research, it is demonstrated that for the chosen dataset, usage of SVM-LRU caching with intra-clustering for pre-fetching technique results in better HR and BHR compared to other popular web caching techniques.

As future enhancement, various other datasets may be experimented for observing the consistency of the results achieved. Also other pre-fetching techniques may be used in place of clustering technique and their efficiency in terms of HR and BHR can be compared.

REFERENCES

1. Datta A., K. Dutta, H. Thomas and D. VanderMeer, 2003. World wide wait: a study of Internet scalability and cache-based approaches to alleviate it. Management Science, 49(10): 1425-1444.
2. Markatos, E. and C. Chronaki, 1998. A top-10 approach to pre-fetching on the web. Proceedings of INET '98, Geneva, Switzerland.
3. Bestavros, A., 1995. Using speculation to reduce server load and service time on the www. Proceedings of the 4th ACM International Conference on Information and Knowledge Management, Baltimore, USA.
4. Padmanabhan, V. and J.C. Mogul, 1996. Using predictive pre-fetching to improve World Wide Web latency. Proceedings of the ACM SIGCOMM'96 Conference, Palo Alto, USA.
5. Zukerman, I., D.W. Albrecht and A.E. Nicholson, 1999. Predicting users' requests on the www. UM '99: Proceedings of the seventh international conference on User Modeling, pp: 275-284.
6. Davison, B.D., 2002. Predicting web actions from html content. Proceedings of the 13th ACM Conference on Hypertext and Hypermedia, College Park, USA.
7. Ibrahim, T. and C. Xu, 2000. Neural nets based predictive pre-fetching to tolerate www latency. Proceedings of the 20th IEEE International Conference on Distributed Computing Systems, Taipei, Taiwan.
8. Fan, L., P. Cao, W. Lin and Q. Jacobson, 1999. Web pre-fetching between low bandwidth clients and proxies: Potential and performance, Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pp: 178-187.
9. Palpanas, T. and A. Mendelzon, 1999. Web pre-fetching using partial match prediction, Proceedings of the 4th International Web Caching Workshop, San Diego, USA.
10. Chen, X. and X. Zhang, 2002. Popularity-based PPM: An effective web pre-fetching technique for high accuracy and low storage, Proceedings of the 2002 International Conference on Parallel Processing, Vancouver, Canada.
11. Nanopoulos, A., D. Katsaros and Y. Manopoulos, 2001. Effective prediction of web-user accesses: A data mining approach. Proceedings of the Workshop on Mining Log Data across All Customer Touch points, San Francisco, USA.
12. Bonino, D., F. Como and G. Squillero, 2003. A real-time evolutionary algorithm for web prediction. Proceedings of the International Conference on Web Intelligence, Halifax, Canada.
13. Mookherjee, V.S. and Y. Tan, 2002. Analysis of a least recently used cache management policy for web browsers, Operations Research, 50(2): 345-357.
14. Koskela, T., J. Heikkonen and K. Kaski, 2003. Web cache optimization with nonlinear model using object feature, Computer Networks journal, Elsevier, 43(6): 805-817.
15. Cobb, J. and H. ElAarag, 2008. Web proxy cache replacement scheme based on back-propagation neural network. Journal of System and Software, 81(9): 1539-1558.
16. Cobb, J. and H. ElAarag, 2006. Training and simulation of neural networks for web proxy cache replacement. Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunications Systems (SPECTS2006), Summer Simulation Multi-conference, Calgary, Canada, pp: 279-298.
17. Calzarossa, V.G., 2003. A Fuzzy Algorithm for Web Caching. Simulation Series Journal, 35(4): 630-636.
18. Farhan, 2007. Intelligent Web Caching Architecture. Master thesis, Faculty of Computer Science and Information System, UTM University, Johor, Malaysia.
19. Qiang Yang and Zhen Zhang, 2001. Model Based Predictive Pre-fetching. Proceedings of IEEE International Workshop on Database and Expert Systems Applications.
20. Craswell, N., D. Hawking and S.E. Robertson, 2001. Effective Site Finding Using Link Anchor Information. Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval.
21. Xu Cheng-Zhong and Tamer I. Ibrahim, 2004. A Keyword Based Semantic Pre-fetching Approach in Internet News Services. IEEE Transactions on Knowledge and Data Engineering, 16(5): 601 -611.
22. Pons Alexander P., 2006. Object Pre-fetching Using Semantic Links. ACM SIGMIS Database, 37(1): 97-109.

23. Georgakis, A. and H. Li, 2006. User behavior modeling and content based speculative web page pre-fetching, Data and Knowledge Engineering - Elsevier, 59: 770-788.
24. Baskaran, K.R. and C. Kalaiarasan, 2014. Pre-eminence of Combined Web Pre-fetching and Web Caching Based on Machine Learning Technique, Arabian Journal for Science and Engineering, 39(11): 7895-7906.
25. Baskaran, K.R. and C. Kalaiarasan, 2016. Improved Performance by Combining Web Pre-fetching Using Clustering with Web Caching Based on SVM Learning Method, International Journal of Computers Communications & Control, 11(2): 167-178.