International Journal of Advance and Applied Research

www.ijaar.co.in

ISSN – 2347-7075 **Impact Factor – 8.141**

Peer Reviewed **Bi-Monthly**

Vol. 6 No. 18 **March - April - 2025**

Machine Learning-Based Optimization of Web Caching: A Support Vector Machine Model

Dr. H. B. Patelpaik

Assistant Professor,

Lokmanya Tilak Mahavidyalaya, Wani. 445304

Corresponding Author – Dr. H. B. Patelpaik

DOI - 10.5281/zenodo.14784834

Abstract:

In the era of information technology, the Internet serves as a critical medium for accessing information globally. The World Wide Web (WWW) facilitates a diverse range of Internet-based services, including e-commerce, online banking, entertainment, education, and e-governance. However, the exponential growth in web applications has led to a substantial increase in network traffic, causing congestion and elevating server loads. This, in turn, results in higher response times, thereby negatively impacting user experience. Web caching has emerged as an effective solution to mitigate latency issues by storing frequently accessed web objects closer to end users. Traditional caching strategies, such as Least Recently Used (LRU), Least Frequently Used (LFU), SIZE, GD-Size, and GDSF, have been widely implemented to enhance web system performance. However, recent advancements in machine learning have significantly improved conventional web proxy caching policies. Support Vector Machine (SVM), a robust supervised machine learning algorithm, is extensively utilized for both classification and regression tasks. By integrating conventional caching policies with SVM-based predictive models, intelligent caching approaches have been developed. These models are evaluated using trace-driven simulations, and their performance is systematically compared with traditional web proxy caching techniques. The empirical findings indicate that SVM-enhanced caching strategies yield substantial performance improvements, demonstrating the efficacy of machine learning in optimizing web caching systems.

Keyword: -WWW, Internet, Web caching, Machine learning, HR, BHR

Introduction:

World Wide Web contains vast amount of information almost on every subject such as business, sports, medical science, banking, shopping, education, environment, defense, etc. This information is available in the form of variety of web objects like text pages, digital images, audios, videos, web applications, etc. Users can access these contents from any part of the world over the internet using their devices such as computers, laptops, and cell phones. The Increasing popularity of WWW has introduced new issues such as Internet traffic, bandwidth consumption thereby leading to latency in service being provided by the application servers. Due to technological advances this huge traffic can lead to significant delays in accessing objects on the web.

Caching of objects in the WWW is widely used technique to reduce network traffic, latency and server load. Caching stores the copies of objects relatively closer to the user. Caching plays a important role to increase the performance of web sites. Caching techniques such as LRU, LFU, GDS, and GDSF etc. are available to caching the objects.

The web objects can be cached at various places at WWW: at the client browser at or near the server (reverse proxy) to reduce the server load, or at a proxy server. When the cache is full and the proxy needs to cache a new object, it has to decide which object to evict from the cache to accommodate the new object. The techniques used for the removal decision is referred to as the replacement policy.

Traditional caching policies, such as Least Frequently Used (LFU), Least Recently Used (LRU), SIZE, and Greedy-Dual-Size-Frequency (GDSF), which are not always perform optimally when applied to World Wide Web (WWW) traffic. This is due to several key factors:

- **Fixed-Size Page Assumption**: Most of caches deal with fixed-size pages in memory system, so the size of the page does not play any role in the replacement policy.
- **Cost of Cache Misses**: The impact of missing a web object depends on multiple factors, such as the physical distance between the proxy and the original server, available bandwidth, and the object's size. Traditional caching policies do not consider these variables.
- **Frequent Web Object Updates**: Web objects are frequently updated, requiring caching policies to account for expiration periods. Memory systems typically do not incorporate expiration constraints for cached pages.
- **Popularity Consideration**: Web object popularity plays a crucial role in optimizing performance metrics. However, traditional memory-based caching policies do not inherently consider popularity trends.
- **Risk of Serving Stale Data**: Without proper proxy updates, a client may end up accessing outdated information, leading to inconsistencies in web content delivery.
- **Increased Latency on Cache Misses**: When a cache miss occurs, additional processing at the proxy server can introduce delays, further increasing access latency.

To overcome these limitations, machine learning techniques have been integrated with traditional caching strategies to enhance web proxy caching performance. Support Vector Machine (SVM), a powerful supervised learning algorithm, is widely used for both classification and regression tasks. By leveraging machine learning, caching policies can adapt dynamically to web traffic patterns, improving efficiency and reducing latency.

Web Caching:

Web caching is one of the most successful solutions for improving the performance of Web-based systems. In Web caching, the popular Web objects that are likely to be used in the near future are stored on devices closer to the Web user such as client's machine or proxy server. Thus, Web caching has three attractive advantages to Web users. Web caching decreases user perceived latency, reduces network bandwidth usage and reduces load on the origin servers. [1, 2] Typically, a Web cache is located in a browser, proxy server and/or origin server as shown in Fig. 1. The browser cache is located in the client machine. At the origin server, Web pages can be stored in a server-side cache for reducing the redundant computations and the server load.

The proxy cache is found in the proxy server, which is located between the client machines and origin server. It works on the same principle as the browser cache, but on a much larger scale. Unlike the browser cache which deals with only a single user, the proxy server serves hundreds or thousands of users in the same way. As shown in Fig. 1, when a request is received, the proxy server checks its cache. If the object is available, the proxy server sends the object to the client. If the object is not available, or it has expired, the proxy server will request the object from the origin server and send it to the client. The requested object will be stored in the proxy's local cache for future requests.

Web proxy caching is widely utilized by computer network administrators, technology providers, and businesses to reduce both user delays and Internet congestion [3, 4]

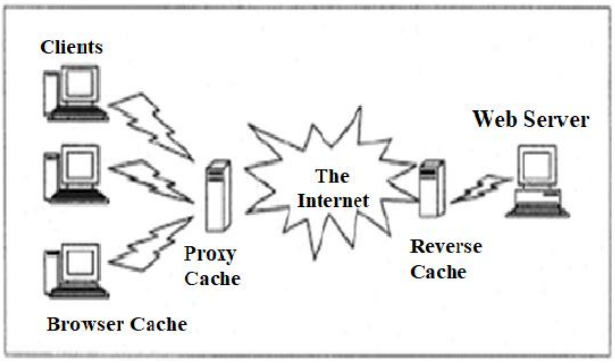


Fig 1: Web object caching position

Cache Replacement Strategies: Cache replacement policies plays an important role in improving Web Caching algorithms. Following are the traditional cache replacement algorithm

- LRU: The least recently used objects are removed first.
- LFU: The least frequently used are removed first.
- SIZE: Big objects are removed first.
- GD-Size: Big objects are removed first.
- GDSF: Extension of GDS by integrating the frequency factor.

Performance Measures:

There are standard metrics available for measuring the efficiency and the performance of the web caching algorithms.

Hit Ratio (HR):- Percentage of requests that can be satisfied by the cache.

Byte Hit Ratio (BHR):- Number of bytes stratified by the cache as a fraction of total bytes requested by the user.

$$HR=\frac{\sum_{i=1}^{N}\delta}{N} \quad \text{OR}$$

Cache hit ratio = [Cache Hits / (Cache Hits + Cache Misses)] x 100 %

$$BHR = \frac{\sum_{i=1}^{N} bi\delta i}{\sum_{i=1}^{N} bi}$$

Numerous studies have been conducted on traditional cache replacement algorithms, demonstrating significant improvements in web server performance. Using a sample dataset, various algorithms such as n-gram, GDSF, GD-Size, LRU, and LFU have been evaluated for efficiency based on Hit Rate (HR) and Byte Hit Rate (BHR) parameters. The performance outcomes are influenced by cache size and the volume of the training dataset. Most research findings indicate that GDSF and LRU exhibit notable performance enhancements.

Techniques of Machine Learning:

Machine learning is a rising technology which supports computers to learn automatically from past data. Machine learning uses different algorithms for **building mathematical models and making predictions using historical data or information.** Now a days, it is used for various tasks such as **image recognition, speech recognition,** medical diagnosis, **email filtering,** stock market trading, **Facebook auto-tagging,** selfdriving cars, product **recommendation**, traffic predictions, online fraud detection etc.

Machine Learning enables to recognize patterns on the basis of existing algorithms and data sets and to develop adequate solution. The machine learning algorithm used the previous examples as inputs, analyzes them, and outputs abstract patterns or rules. Thus, the machine learning mechanisms form the basis for adaptive systems. [1]

Web log files give the information about the behavior of user. With the help of log data, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning techniques are used to implement the aforesaid objectives. Now a day's research is going on to combine traditional cache replacement policies with machine learning techniques which may be called as hybrid techniques. This paper discusses the effect of hybrid techniques in enhancing the performance of the web server. Naïve Bayes, Decision Tree and Support Vector Machine are the three machine learning techniques were taken into consideration and their impact on enhancing the performance were discussed here

Support Vector Machine:

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. It performs classification more accurately and faster than other algorithms. These machine learning algorithms have a wide range of applications such as text classification, Web page classification and bioinformatics application. Hence SVM can be used to produce promising solution for web proxy caching.

In SVM, each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper plane that differentiate the two classes very well. Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line) as shown in Fig.2 below. Support vector machine or SVM algorithm is based on the concept of „decision planes", where hyper planes are used to classify a set of given objects. Pictorial examples of support vector machine algorithm in Fig.2 shows that two sets of data. These datasets can be separated easily with the help of a line, called a decision boundary. There can be several decision boundaries that can divide the data points without any errors. The nearest points from the optimal decision boundary that maximize the distance are called support vectors. [5,6]

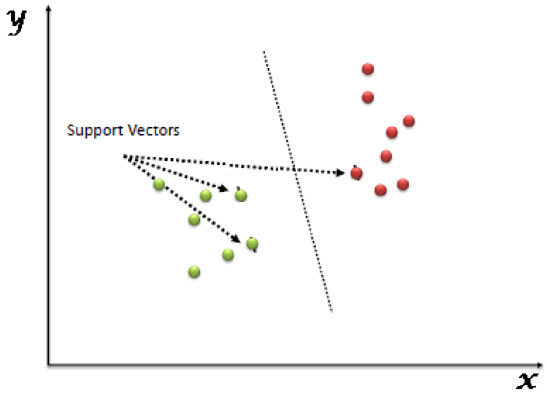


Fig. 2 – SVM-Segregated Hyper plane

Enhancing Web Caching with SVM:

Support Vector Machine algorithm is used in combination with the traditional caching techniques to enhance the performance. This technique is belonging to the category of supervised learning. In supervised learning the knowledge is acquired from the available data set to extract the data pattern. First, we develop the machine learning model of the respective technique. Then we train and test the model. We split the available dataset into two parts, namely training-data-set and testing data set. A supervised learning model analyzes the training data and produces a new knowledge, which is then used for predicting new information. The perfectly trained model will be able to correctly determine the class labels for unseen instances.[7]

In each case a trained classifier has been designed first and then training data set is used as a model to select the appropriate web object from the remaining data set for cache replacement. In case of all these three techniques a hybrid model is used i.e. by combining respective machine learning technique with traditional caching algorithms. The performances of each technique are analyzed against the metrics Hit Ratio (HR) and Byte Hit Ratio (BHR). On overall analysis of the three cases the following interesting results are found which can be found more useful for enhancing the performance of the web Server and proxy server.

- Support Vector Machine algorithm combined with traditional cache replacement algorithms; the performance is considerably improved.
- Cache Size plays an important role in enhancing the performance. The results are found noticeable in case of medium sized caches as compared to small or very large sized cache.

Evaluation Based on Hit Ratio:

The performance of caching policies is measured in terms of Hit Ratio (HR) and Byte Heat Ratio (BHR). When a requested web object is present in the cache, cache-hit occurs and the request is served from the cache. These requests are saved from being reached to the server, which ultimately results in,

i. Reducing the latency as the request need not to reach to the server and is served from the cache itself immediately. Thus, the time required for fetching the web object from server is saved.

ii. Reduced server load, as the requests equal to the number of cache hits are need not be served by the origin server.

iii. Saving network bandwidth, as the web objects for which cache hits occur do not required to be fetched from origin server which saves considerable bandwidth on the network between proxy server and the origin server.

In this section the performance in terms of Hit Ratio and Byte Hit Ratio of cache policies is presented. The performance of the proposed model is measured in the combination of,

i. Accuracy of the machine learning model, and

ii. The Hit Ratio and Byte Hit Ratio achieved

The results of cache simulator showing number of Request Hits and its ratio against total number of requests coming from the clients (Hit Ratio) are shown in table 1.0 and 1.1 respectively.

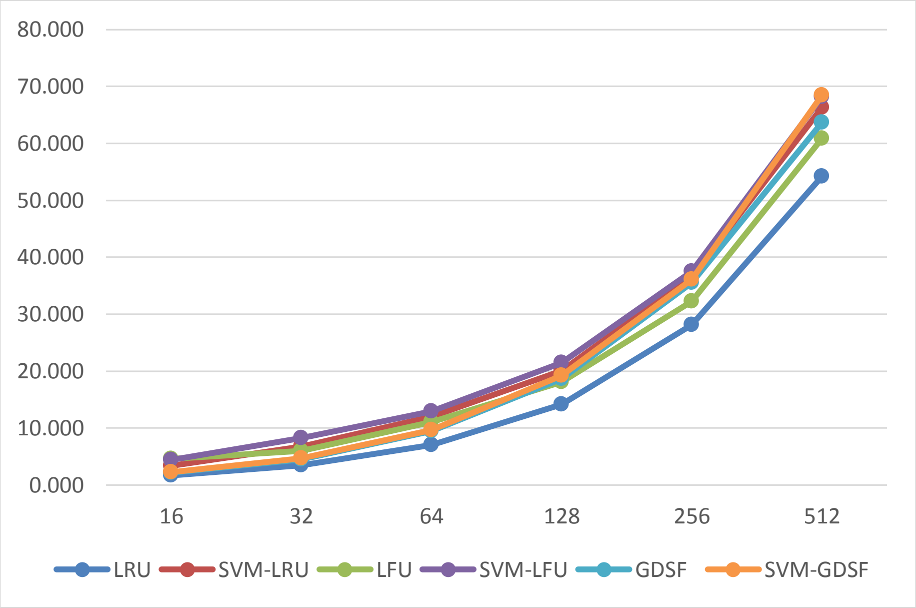
Table No.1.0 Request hits of different caching policies against given input values

ng Policy	Cache Size [MB]						
	6	32	64	128	256	512	1024
	597	3489	088	14137	8158	4178	1610
LRU	341	5770	990	20071	6675	5315	3231
	543	5052	109	18078	2225	0800	4742
LFU	448	3213	973	21454	7500	3104	3843
	162	4554	480	18616	5504	3667	2626
GDSF	247	4661	589	19262	6138	3455	4260

Table No: 1.1 Hit Ratio of Conventional and SVM based Policies

ng Policy	Cache Size [MB]						
	6	32	64	128	256	512	1024
	597	3.489	088	4.137	.158	4.178	1.610
LRU	341	6.770	.990	0.071	.675	6.315	3.231
	543	6.052	.109	8.078	.225	0.800	4.742
LFU	448	8.213	.973	1.454	.500	8.104	3.843
	162	4.554	480	8.616	.504	3.667	2.626
GDSF	247	4.661	689	9.262	.138	8.455	4.260

Graph 1.1 Hit Ratio of Conventional and SVM Based policies



The table 1.0 and table 1.1 above shows the number of request hits and hit ratio of each policy for different cache sizes. The comparison of performance of different cache policies is shown in graph. It is clearly seen that the performance of machine learning based policies against their convectional counterparts is better. On the other hand, graph 1.1 shows that, as the cache size increases the performance of caching policies improves. This is because as the cache size increases the chance of occurrences of cache miss reduces, that means, more and more number of requests are served from cache which in turn reduce server load, save bandwidth and reduce latency

Conclusion and Future Scope:

The application of Machine Learning techniques has become a significant area of research across various fields, including business, manufacturing, financial analysis, education, and sports. Web performance optimization is no exception. With the increasing reliance on web applications and social networking, the demand for internet resources especially bandwidth has grown substantially. However, bandwidth expansion has its limitations, making it essential to explore new techniques for reducing network traffic.

Web caching plays a crucial role in minimizing internet traffic, but traditional caching techniques have inherent limitations. To enhance web caching efficiency, Support Vector Machine (SVM) can be integrated with conventional caching methods. This paper explores how combining SVM with traditional web caching strategies can help reduce access latency and improve overall performance.

SVM learns from web proxy log files to classify objects based on their likelihood of being revisited. Additionally, other intelligent classifiers can be employed to further refine caching policies. Several intelligent web proxy caching approaches can be developed to improve both the hit ratio (the percentage of requested objects found in the cache) and the byte hit ratio (the proportion of data served from the cache relative to total data requested). Experimental results demonstrate a significant improvement in web access time, highlighting the effectiveness of machine learning-enhanced caching strategies.

References:

1. Intelligent Web Caching Using Machine Learning Methods, Sarina Suleman, Sitimariyan Shamsuddin, Ajith Abraham, Shahida Sulaiman
2. A survey of web caching and prefetching. Ali, Waleed, Siti Mariyam Shamsuddin, and Abdul Samad Ismail. *Int. J. Advance. Soft Comput. Appl* 3.1 (2011): 18-44.
3. An admission-control technique for delay reduction in proxy caching, C.C. Kaya, G. Zhang, Y. Tan, V.S. Mookerjee, Decision Support Systems 46 (2009) 594–603
4. Web cache optimization with nonlinear model using object features, T. Koskela, J. Heikkonen, K. Kaski, Computer Networks 43 (2003) 805–817.
5. Analysis of various techniques for improving Web performance, Sofi, Ayaz Ahmad, and Atul Garg. (2015).
6. Performance Improvement of Least-Recently-Used Policy in Web Proxy Cache Replacement Using Supervised Machine Learning, Waleed Ali, Sarina Suleman, Article *in* International Journal of Advances in Soft Computing and its Applications · March 2014
7. IntellCache: An Intelligent Web Caching Scheme for Multimedia Contents, Nishat Tasnim Niloy, Md. Shariful Islam, Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2021.