

Performance Improvement of Web Proxy Cache Replacement using Intelligent Greedy-Dual Approaches

Waleed Ali

Department of Information Technology
Faculty of Computing and Information Technology, King Abdulaziz University
Rabigh, Kingdom of Saudi Arabia

Abstract—This paper reports on how intelligent Greedy-Dual approaches based on supervised machine learning were used to improve the web proxy caching performance. The proposed intelligent Greedy-Dual approaches predict the significant web objects' demand for web proxy caching using Naïve Bayes (NB), decision tree (C4.5), or support vector machine (SVM) classifiers. Accordingly, the proposed intelligent Greedy-Dual approaches effectively make the cache replacement decision based on the trained classifiers. The trace-driven simulation results indicated that in terms of byte hit ratio and/or hit ratio, the performance of each of the conventional Greedy-Dual-Size-Frequency (GDSF) and Greedy-Dual-Size (GDS) was noticeably enhanced by applying the proposed Greedy-Dual approaches on five real datasets.

Keywords—Cache replacement; Greedy-Dual approaches; machine learning; proxy

I. INTRODUCTION

Internet performance can be improved by several approaches, any one of which may not always be the best method, due to practical issues such as network infrastructure, environment, and cost of hardware [1]. The second and the most popular approach is a web caching technique [1], [2], which decreases the network load by providing the requested web content from local storage. In a similar manner to caching in the cache memory to enhance CPU performance, web caching stores some web objects in anticipation of future requests, to enhance Web-based systems.

Basically, the implementation of web caching is done in three levels: client machine, proxy server and/or origin server. However, it is considered that the most significant caching approach is web proxy caching [2]-[7] which is used to save the networks' bandwidth, reduce Internet network traffic and decrease user-perceived latency.

In some situations, the proxy cache buffer is full of the stored web objects and a cache replacement policy is executed to provide enough space for the new incoming objects. The proxy cache replacement policy is responsible for removing unwanted web objects which may cause proxy cache pollution and poor performance.

Greedy-Dual-Size-Frequency (GDSF) and Greedy-Dual-Size (GDS) are two of the most commonly used web pages caching strategies, which are applied at proxy server. In GDS

and GDSF, the replacement cache decision is made based on mathematical equations combining a few important features of the object. Higher priority is given by GDS and GDSF to small web objects compared with large objects. Thus, the hit ratio is maximized, but at the expense of the byte hit ratio. Since web users' interests change depending on rapid changes in a web environment, smart and adaptive approaches are required to contribute to the web caching and replacement decisions.

II. SUMMARY OF CONTRIBUTIONS

Least-Frequently-Used-Dynamic-Aging (LFU-DA) and Least-Recently-Used (LRU) were enhanced using supervised machine learning in previous works [7], [8], respectively. However, the hit ratio measure achieved by the intelligent LRU and LFU-DA approaches were not good enough compared to GDS and GDSF because neither the size nor the retrieving time of web pages was considered in these approaches. In this paper it is shown how intelligent machine learning classifiers are effectively utilized in the GDS and GDSF in order to obtain optimal and intelligent Greedy-Dual approaches that can perform better in terms of both bytes hit ratio and hit ratio.

GDS is combined with intelligent machine learning classifiers to produce novel smart GDS caching methods (such as SVM-GDS, C4.5-GDS, and NB-GDS) with better performance. In the proposed intelligent GDS caching approaches, the frequency factor in the conventional GDS policy is replaced with the probability (computed by either the trained C4.5, SVM or NB classifier) of re-accessing the object soon.

In addition, C4.5, SVM or NB is incorporated with GDSF to improve the low byte hit ratio. The subsequent proposed replacement approaches are called C4.5-GDSF, SVM-GDSF and NB-GDSF. In the proposed intelligent GDSF approaches, the value of the object class (either one or zero) predicted by the trained classifier is added in the conventional GDSF in order to assign a higher priority to the web objects that are likely to be revisited soon.

The relative performances of the proposed intelligent Greedy-Dual approaches are then comprehensively discussed and compared with the most common and more relevant intelligent cache replacement methods.

The remainder of this paper is structured as follows. Section III describes the background of web proxy replacement and caching. Supervised machine learning is also presented briefly in subsection B while the current intelligent web cache replacement techniques are summarized in subsection C. Section IV presents the methodology of the proposed intelligent Greedy-Dual algorithms. The proposed approaches are evaluated and compared with other conventional and intelligent cache replacement techniques in Section V. Finally, Section VI concludes the work proposed in this study and suggests future work arising from this paper.

III. BACKGROUND AND RELATED WORK

A. Web Proxy Cache Replacement

The web proxy caching is a useful technique that plays an essential role in improving the performance of Web-based systems in terms of minimizing the utilization of network bandwidth, decreasing user-perceived delays and reducing loads on the original servers.

Three popular aspects have high impact on web proxy caching, which are cache consistency, cache pre-fetching, and cache replacement [1], [3], [4]. However, the powerful cache replacement method is essential and can make the greatest contribution in enhancing the caching performance [5]-[10].

When the proxy cache becomes full of web objects, a replacement strategy is basically used to manipulate the contents of the cache to provide sufficient space for incoming objects. The primary objective of the ideal cache replacement policy is to eliminate the undesired objects, to provide the best utilization of the proxy cache. Hence, cache hit rates can be improved, and loads on the server can be reduced.

A Greedy-Dual-Size (GDS) policy is suggested by [11] to lessen the cache pollution issues faced by the SIZE policy. In addition to the size factor, the cost of retrieving a web object from the server and the aging factor are combined with the key value assigned by GDS for each object available in the proxy cache. As the proxy cache is fully occupied, the web object that has the lowest key value is removed to provide enough place to the new demanded objects. The GDS policy uses (1) to compute $K(g)$, which represents the caching priority of object g visited by a web user.

$$K(g) = L + \frac{C(g)}{S(g)} \quad (1)$$

Where $S(g)$ is the size of g ; $C(g)$ is the fetching cost of g from its origin server; and L is an aging factor, which has the zero as the initial value and is then adjusted to the caching priority of the last replaced object.

When object g is requested again, $K(g)$ is modified based on the updated L value. Hence, the objects visited recently have larger caching priority values. The GDS policy obtains a much better hit ratio compared with other conventional replacement methods. However, the GDS approach still suffers from a low byte hit ratio [11]. Therefore, [12] suggested

an improvement on GDS by integrating the visit frequency $F(g)$ into the replacement decision, to produce Greedy-Dual-Size-Frequency (GDSF), as shown in (2). GDSF accomplishes a higher hit ratio compared to other cache replacement methods. However, although GDSF obtains a higher byte hit ratio than GDS, GDSF still performs minimal byte hit ratio compared to the other conventional replacement methods [12].

$$K(g) = L + F(g) * \frac{C(g)}{S(g)} \quad (2)$$

B. Supervised Machine Learning

The supervised learning algorithm works on the training dataset to generate a classifier that has the ability to predicting the correct class for the known dataset (testing dataset). This section concentrates on three popular machine learning algorithms: decision tree (C4.5), support vector machine (SVM) and Naïve Bayes classifier (NB), which have been successfully applied in many applications [13]-[17].

In the decision tree, a feature in the training instance is represented by a node, while each tree branch has a value, which can be predicted by that node. The C4.5 developed by [17] is the most commonly used algorithm to generate a decision tree for classification purposes. The C4.5 is constructed based on a top-down recursive approach to generate the decision tree. All of the training instances are initially at the tree root. The C4.5 then uses an impurity function in order to split the training instances recursively. The partitioning process is then repeated until all instances for a given node belong to the same class.

A support vector machine, which is a discriminative model, aims to achieve an optimal hyperplane which categorizes new instances by generating the maximal likely distance between the separating hyperplane and the instances in order to decrease the upper bound on the predictable generalization error. In the SVM training, support vectors closer to the separating hyperplane are obtained from the dataset to represent the most valuable instances used for classification. In addition to linear classification, SVMs can be used to solve other non-linear classification problems by selecting the appropriate kernel function to convert the instances into high-dimensional spaces.

One of the simplest Bayesian networks is the Naive Bayes network (NB), which is represented as a directed acyclic graph in which the class label is represented by the single parent and the features are represented by some children. NB supposes that no correlation exists between the features and that, given the class label, all the features are conditionally independent. The conditional probabilities $\Pr(A_i = a_i | C = c_j)$ and the prior probabilities $\Pr(C = c_j)$ are computed in the training phase. Formula (3) is then used in order to predict the class of a test example.

$$c = \arg \max_{c_j} \Pr(C = c_j) \prod_{i=1}^{|A|} \Pr(A_i = a_i | C = c_j) \quad (3)$$

TABLE I. SUMMARY OF THE EXISTING INTELLIGENT WEB CACHE REPLACEMENT TECHNIQUES

Machine Learning Used	Existing Works	Based on	Cache Location	Data Used for Evaluation
SVM	SVM-DA [7]	LFU-DA	Proxy	The IRCache network's proxy logs files
	SVM-LRU [8]	LRU	Proxy	The IRCache network's proxy logs files
Decision Tree	C4.5-DA [7]	LFU-DA	Proxy	The IRCache network's proxy logs files
	C4.5-LRU [8]	LRU	Proxy	The IRCache network's proxy logs files
	J48-C-LRU [22]	LRU	Proxy	The IRCache network's proxy logs files
	CART, MARS, RF and TN in web caching [23]	LRU	Client and Sever	- Cunha of Boston University 's Web client traces - E-Learning @UTM Web server
Naïve Bayes Classifier	NB-DA [7]	LFU-DA	Proxy	The IRCache network's proxy logs files
	NB-LRU [8]	LRU	Proxy	The IRCache network's proxy logs files
ANN	NNPCR [24] and NNPCR-2 [6]	LFU-DA	Proxy	The IRCache network's proxy logs files
	BP and PSO in Web caching [25]	None	Server	Cunha of Boston University 's Web client traces
	LRU-C [9]	LRU	Server	Finnish University and Research Network access's logs file
ANFIS	ICWCS [26]	LRU	Client	Cunha of Boston University 's Web client traces
Logistic Regression	LRU-C and LRU-M [27]	LRU	Proxy	The IRCache network's proxy logs files just for one day
	Logistic regression in an adaptive web cache [28]	None	Server	Server logs files's Internet Traffic Archive

C. Related Works on Intelligent Web Cache Replacement Techniques

Several intelligent methods have been explored as alternative solutions to enhance the performance of traditional approaches of proxy cache replacement. The intelligent proxy cache replacement methods have been developed by using supervised machine learning techniques (see Table I), fuzzy systems [18], or evolutionary algorithms [19]-[21]. The existing intelligent web cache replacement techniques based on the supervised machine learning are considered as the most commonly used, effective and adaptive approaches, as summarized in Table I.

By examining the existing works cited in Table I, it can be concluded that two intelligent replacement paradigms are dominant in the existing intelligent web cache replacement techniques. A supervised machine learning technique is utilized independently in the proxy cache replacement or incorporated with one of the conventional replacement policies such as LFU-DA or LRU. The object size and cost are not considered in the replacement decision with these paradigms.

Unlike the previous works, the proposed intelligent Greedy-Dual approaches can remarkably enhance the byte hit ratio of the conventional GDSF and GDS. Besides, they utilize the advantages of GDS and GDSF in terms of high hit ratio. In other words, intelligent machine learning classifiers are effectively utilized into the GDS and GDSF in order to obtain optimal intelligent Greedy-Dual approaches that can achieve good performance in both the hit ratio and the byte hit ratio.

IV. METHODOLOGY

A methodology for enhancing web proxy cache replacement using intelligent Greedy-Dual approaches is explained in this section. The methodology involves two phases: training of supervised machine learning classifiers, and then integrating the trained classifiers into the web proxy cache replacement.

A. Training of Supervised Machine Learning Classifiers

In order to effectively predict the desired web object, C4.5, SVM and NB classifiers are trained with training data prepared based on users' requests recorded in the web proxy logs file. Some features of the training dataset are extracted from the web proxy logs file immediately, while other features are prepared using equations, as shown in Table II. The target output for each request is also prepared from the proxy logs file, based on the forward-looking sliding window (SWL) as shown in (4).

$$y = \begin{cases} 1, & \text{if the object would be revisited within forward SWL} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

As can be observed, the input features are based on the past information of objects requests within the backward-looking sliding window to expect whether these objects would be revisited soon or not within the forward-looking sliding window.

TABLE II. THE FEATURES PREPARATION OF TRAINING DATASET

Feature Name	Description	How to Prepare
SWL-based Recency	Recency of visiting a object withing backward-looking sliding window	$x_1 = \begin{cases} \text{Max}(SWL, \Delta T) & , \text{if object } g \\ & \text{was requested before} \\ SWL & , \text{otherwise} \end{cases}$ where ΔT is the time in seconds since object g was last request , and SWL is sliding window length.
Frequency	Visits Frequency of a object	Number of requests for a web object in proxy logs file
SWL-based Frequency	Visits Frequency of a object within backward-looking sliding window	$x_{3,i} = \begin{cases} x_{3,i-1} + 1 & , \text{if } \Delta T \leq SWL \\ x_{3,i-1} & , \text{otherwise} \end{cases}$
Retrieval time	fetching time of a object in milliseconds	extracted from elapsed time field of log entry in the proxy logs file
Size	Size of object in bytes	extracted from size field of log entry in the proxy logs file
Type	Type of web object	1 for HTML, 2 for image, 3 for audio, 4 for video, 5 for application and zero for others.

When the proxy dataset is preprocessed well, C4.5, SVM and NB can be trained using the prepared dataset for web object classification. The training phase aims to train C4.5, SVM and NB classifiers to predict the web object class requested by the user, either as objects to be revisited soon or not. Consequently, the classification information is utilized with the cache replacement decision to enhance the web proxy caching performance.

B. Proposed Intelligent Greedy-Dual Approaches

As NB, C4.5 and SVM are correctly trained to classify proxy cache contents, as discussed earlier; a web proxy cache replacement strategy can utilize NB, C4.5 or SVM classifiers for managing the contents of the web proxy cache. As shown in Fig. 1, when a web user visits object g , the cache manager searches for object g in the proxy cache. Whether a cache hit or miss has occurred, intelligent Greedy-Dual approaches are used to compute or update the caching priority, $K(g)$, of g . The desired features of g , as shown in Table II, are collected and utilized as inputs for the classification algorithm that can classify object g as an object that would be revisited again or not. Thus, the classification decision is incorporated into the GDS or GDSF cache replacement approach for updating the priority of g . Then, g is reordered and located depending on the new priority of g in the cache list. Consequently, the proposed intelligent GDS and GDSF can identify and remove the unwanted web objects with the lowest priority for replacement.

In the proposed intelligent GDS approaches, classification information produced by C4.5, SVM or NB classifier is combined with the conventional GDS to enhance the byte hit ratio. The suggested intelligent GDS approaches are so-called NB-GDS, C4.5-GDS and SVM-GDS. In the proposed intelligent GDS approaches, a NB, C4.5 or SVM classifier is used to compute the probability, $Pr(g)$, of revisiting object g in the near future. Each time a user visits an object g , the accumulated $Pr(g)$, i.e., $W(g) = \sum Pr(g)$, is combined with the caching priority $K(g)$ using (5).

$$K(g) = L + W(g) * \frac{C(g)}{S(g)} \tag{5}$$

In addition to the intelligent GDS, the traditional GDSF is extended based on a NB, C4.5 or SVM classifier to enhance the low byte hit ratio. Therefore, the proposed NB-GDSF, C4.5-GDSF and SVM-GDSF are produced as alternative approaches to the traditional GDSF web proxy cache replacement method.

In the proposed intelligent GDSF approaches, the trained NB, C4.5 or SVM classifier is applied for the prediction of the web objects' class (one or zero) requested by the web user. The class label is then included as an additional weight into GDSF to provide higher priority to the preferred objects, which will be revisited sometime in future even if the preferred objects are large. When a web user visits g , the intelligent GDSF uses (6) to assign the caching priority, $K(g)$, of object g . Hence, based on its priority, g is relocated in the proxy cache.

$$K(g) = L + F(g) * \frac{C(g)}{S(g)} + W(g) \tag{6}$$

Where $W(g)$ is either one or zero, which represents the class of object g obtained using the NB, C4.5 or SVM classifier.

The rationale behind the proposed intelligent Greedy-Dual approaches is explained as follows. The conventional GDS and GDSF give greater priority to small web objects, which are removed first from the proxy cache. Thus, the hit ratio is maximized by the conventional GDS and GDSF but at the expense of the byte hit ratio. Instead of that, the suggested intelligent Greedy-Dual approaches can predict either the class value or probability of the preferred objects, which would be re-accessed soon using SVM, NB and C4.5 classifiers. Accordingly, the class information is successfully integrated with the storing priority of the web object. In other words, the priority values of those preferred objects can be enhanced using a SVM, NB or C4.5 classifier, regardless of their size and visits frequency. Thus, the proposed intelligent Greedy-Dual approaches can outstandingly enhance the byte hit ratio of the conventional GDS and GDSF. In addition, the superior hit ratio of the conventional GDS and GDSF can be maintained in the intelligent Greedy-Dual approaches.

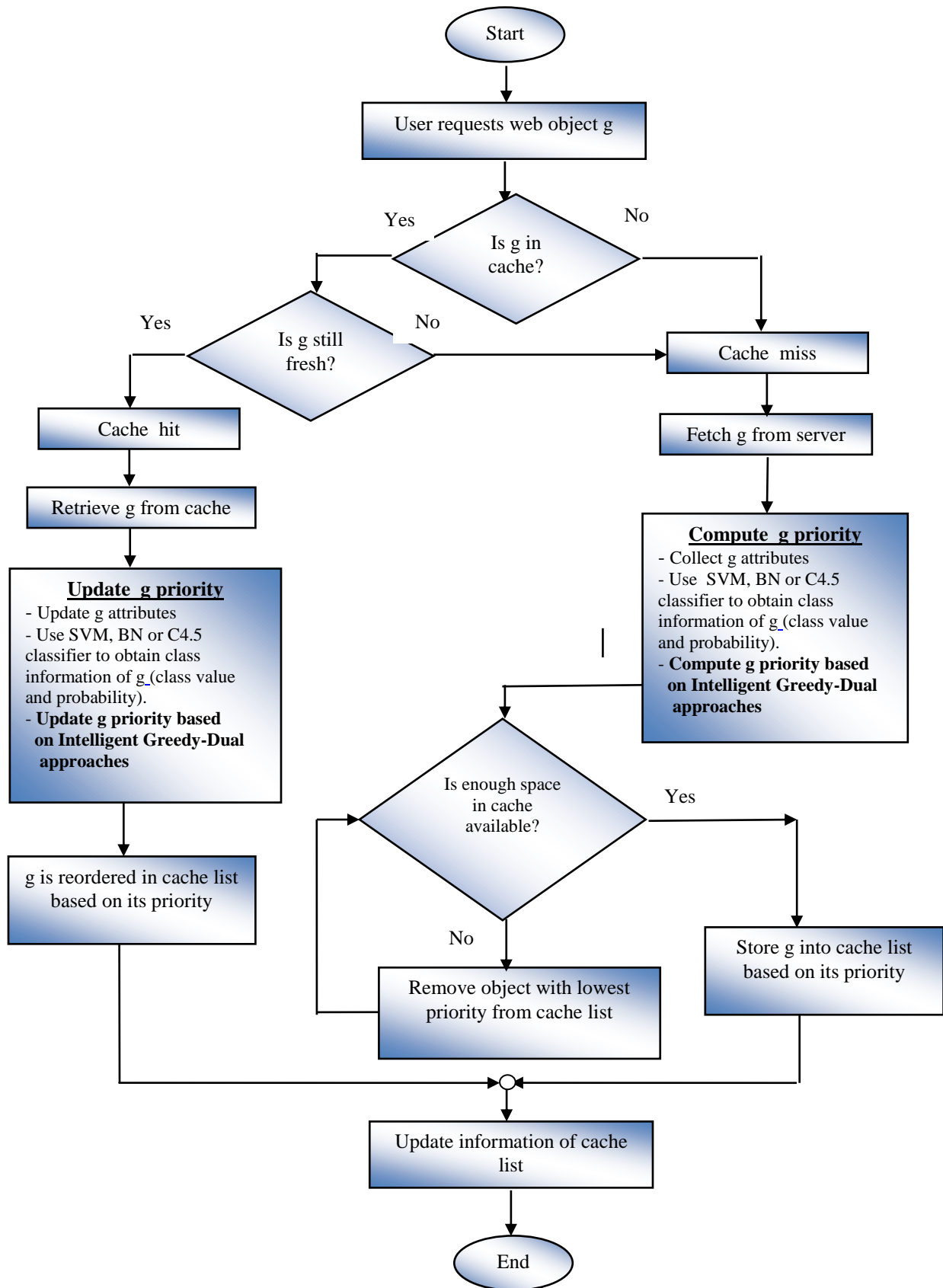


Fig. 1. A methodology for enhancing web proxy cache replacement using intelligent greedy-dual approaches.

V. RESULTS AND DISCUSSION

A. Data Collection

The proxy log files used in this study were obtained from five proxy servers (BO2, NY, UC, SV and SD) from the IRTCache network [29] that are located in the United States over a period of fifteen days. C4.5, SVM and NB classifiers were trained based on the data collected in the first day, while the remaining data of the two weeks were used to evaluate the suggested intelligent Greedy-Dual method against existing works.

B. Improvement Ratio of Hit and Byte Hit Ratio

In this study, a WebTraff [30] simulator was adjusted to simulate and evaluate the effectiveness of the performance of the proposed intelligent Greedy-Dual approaches against various existing web cache replacement policies.

The most popular measures used to verify and evaluate the performance of proxy cache replacement are hit ratio (HR) and byte hit ratio (BHR), which are related with the number of user's requests and bytes served by the proxy cache instead of the original server. Due to space limitations, (7) is used to calculate the average improvement ratios (*IRs*) of conventional method (*CM*) in terms of the HR and BHR obtained by the proposed method (*PM*), i.e., the intelligent GDS and GDSF against conventional GDS and GDSF.

$$IR = \frac{(PM - CM)}{CM} \times 100 (\%) \quad (7)$$

For the five datasets, Table III summarizes the average *IRs* performed by intelligent GDS approaches over conventional GDS for each particular cache size. The averages *IRs* were significantly influenced when the proxy cache size increased. More particularly, the impact of the performance of a replacement policy for the small cache was noticed clearly, since the replacement process occurred frequently.

For HR, the results show that SVM-GDS, NB-GDS and C4.5-GDS improved the HR of GDS with average *IRs* by up to 17.42%, 22.45% and 18.79% respectively, as shown in Table III. For the average *IRs* of the BHR, the BHR of the GDS was significantly enhanced by SVM-GDS, NB-GDS and C4.5-GDS, by up to 57.61%, 229.14% and 85.65%, respectively. This was mainly due to the capability of intelligent GDS approaches to intelligently remove the correct objects from the proxy cache. By contrast, the low BHR of the conventional GDS was expected, due to the GDS's weighting toward smaller objects, even if the smaller objects are not preferred.

From Table III, it can also be seen that the HR of C4.5-GDS was almost the same as the HR of SVM-GDS, but slightly lower than that of NB-GDS. In terms of the BHR, NB-GDS accomplished the best BHR, while SVM-GDS attained the worst BHR compared to the BHRs of NB-GDS and C4.5-GDS. This was due to the fact that NB-GDS gave more

accurate probabilities or scores to the preferred objects, either small or large objects. This contributed greatly to obtaining a good HR and a much better BHR from NB-GDS than from the others.

The average *IRs* achieved by the intelligent GDSF methods are also presented in Table III. SVM-GDSF, NB-GDSF and C4.5-GDSF accomplished good HRs but these were slightly inferior to the HR of the conventional GDSF. In the worst case, SVM-GDSF, NB-GDSF and C4.5-GDSF lost 7.29%, 9.43% and 7.4% respectively from the HR of GDSF. However, the BHR of GDSF was significantly enhanced by SVM-GDSF, GDSF-NB and C4.5-GDSF and increased by 407.49%, 380.55%, and 308.08%, respectively. This enhancement was obtained because the GDSF tends to cache many of the small objects in the proxy cache to increase the HR, but at the expense of BHR.

Table III shows also that C4.5-GDSF and SVM-GDSF achieved slightly higher HRs than the HR of NB-GDSF, while NB-GDSF and SVM-GDSF achieved better BHRs compared to the BHR of C4.5-GDSF. This meant that the best balance between the HR and BHR was achieved by SVM-GDSF.

C. Overall Comparison and Discussion

As shown in the previous section, the proposed NB-GDS and SVM-GDSF achieved a more competitive HR and better BHR. Thus, NB-GDS and SVM-GDSF were selected to be used in the overall comparison. The proposed NB-GDS and SVM-GDSF approaches were compared with the most common cache replacement methods used in squid software such as LRU, GD, GDSF and LFU-DA [24], [6]. In addition, NB-GDS and SVM-GDSF were compared with other existing intelligent proxy cache replacement methods, such as NNPCR-2 [6], SVM-LRU [8], and SVM-DA [7].

In terms of the HR, Fig. 2 clearly indicates that SVM-LRU, NB-GDS and SVM-DA improved the performance of LRU, GDS and LFU-DA, respectively on the five proxy datasets. Conversely, the HR of SVM-GDSF was comparatively or somewhat worse than the HR of GDSF. Fig. 2 also demonstrates that the HRs of NB-GDS, SVM-GDSF and SVM-DA were much better than the HR of NNPCR-2, while the HR of SVM-LRU was slightly better than that of NNPCR-2 for most of the proxy datasets. From Fig. 2, it can be concluded that the best HR was achieved by NB-GDS, while the worst HR was given by LRU on all datasets.

In terms of BHR, Fig. 3 demonstrates that, for all proxy datasets, the BHR obtained by GDS and GDSF was much lower than that achieved by LFU-DA, LRU and NNPCR-2. This was expected, since LFU-DA, LRU and NNPCR-2 policies removed objects regardless of their sizes. Furthermore, the BHRs of SVM-DA and SVM-LRU were better than those of LFU-DA, LRU and NNPCR-2 in all proxy datasets with different cache sizes.

It can also be noticed from Fig. 2 and 3 that although GDS and GDSF had a better performance for the HR obtained compared to the others, it can clearly be seen that the BHRs of GDS and GDSF were the worst among all the methods. This is because GDS and GDSF prefer to cache the small and recent objects.

TABLE III. THE AVERAGE IRs ACHIEVED BY INTELLIGENT GDS AND GDSF OVER GDS AND GDSF

Cache Size (MB)	Average IR of HR and BHR Over GDS and GDSF (%)											
	SVM-GDS		SVM-GDSF		NB-GDS		NB-GDSF		C4.5-GDS		C4.5-GDSF	
	HR	BHR	HR	BHR	HR	BHR	HR	BHR	HR	BHR	HR	BHR
1	17.42	26.23	-1.59	16.42	22.45	46.47	-0.86	16.85	18.79	33.96	-1.03	16.34
2	15.87	26.37	-3.13	27.31	18.96	51.38	-4.55	105.22	17.05	35.21	-2.46	20.43
4	10.56	23.95	-3.88	45.61	15.21	96.64	-10.81	255.28	14.52	28.70	-6.07	32.85
8	13.15	41.09	-5.37	155.77	15.25	229.14	-6.97	278.46	13.97	56	-3.27	125.2
16	11.05	48.47	-6.01	123.57	12.92	179.98	-6.47	147.21	11.6	85.65	-4.35	91.98
32	10.09	54.44	-6.50	94.15	11.31	125.3	-7.76	96.59	10.45	81.01	-5.99	76.7
64	9.74	49.08	-6.39	407.49	10.7	202.07	-7.79	380.55	9.95	54.82	-6.18	308.08
128	7.08	57.61	-7.28	125.95	8.77	105.77	-9.43	96.45	7.29	73.1	-7.4	93.04
256	5.11	50.59	-6.96	84.55	7.65	88.36	-8.79	65.61	5.32	59.39	-6.22	54.34
512	2.96	44.78	-5.32	61.66	5.58	75	-8.18	38.09	3.09	59.61	-4.88	34.42
1024	1.95	35.22	-4.13	40.8	5.32	71.87	-6.44	25.29	2.23	51.34	-2.94	19.38
2048	0.64	34.71	-2.50	25.59	4.47	55.11	-5.11	17.18	0.87	54.97	-2.63	8.15
4096	0.28	28.75	-2.49	9.16	4.41	30.82	-4.08	10.23	0.40	29.89	-1.61	4.81
8192	0.15	8.17	-0.37	2.54	3.86	8.66	-2.06	2.40	0.20	8.42	-0.69	1.95
16384	0.03	1.27	-0.05	1.16	3.89	1.56	-0.25	0.24	0.05	1.58	-0.20	0.12
32768	0	0.56	0	0.02	3.87	0.96	0	0.33	0	0.94	-0.02	0.28

Fig. 2 and 3 show that both SVM-GDSF and NB-GDS were significantly improved in terms of BHRs achieved over GDSF and GDS, respectively. It can be concluded that the proposed NB-GDS and SVM-GDSF achieved outstanding HRs and competitive BHRs for most of the proxy datasets.

VI. CONCLUSION AND FUTURE WORK

In this paper, intelligent Greedy-Dual approaches have been suggested to obtain optimal web proxy cache replacement approaches that can achieve good performance in both HR and BHR. To improve the lower byte hit ratios of the conventional GDS and GDSF policies, intelligent machine learning classifiers were combined with these policies to produce novel intelligent GDS and GDSF caching approaches with better performance. The trace-driven simulation results depicted that the intelligent Greedy-Dual approaches noticeably enhanced the performance of the traditional GDS in terms of byte hit ratio and hit ratio. The averages of the IRs of the BHRs obtained by SVM-GDS, NB-GDS and C4.5-GDS over GDS increased by 57.61%, 229.14% and 85.65%, respectively, while the average IRs of the HR increased by 17.42%, 22.45% and 18.79%, respectively. Moreover, the intelligent GDSF approaches significantly improved the performance in terms of

the byte hit ratio of GDSF. The average IRs of the BHRs of SVM-GDSF, NB-GDSF, and C4.5-GDSF were many times greater than the BHRs of GDSF, and increased by 407.49%, 380.55%, and 308.08%, respectively. When the proposed intelligent Greedy-Dual approaches were compared with conventional and other intelligent replacement approaches, it was observed that the proposed NB-GDS achieved the best HR. Furthermore, BHRs of SVM-GDSF and NB-GDS were competitive with the BHRs of LRU and LFU-DA for most proxy datasets.

The proposed intelligent Greedy-Dual approaches can be implemented in real environments such as organizations or universities. For example, the proposed approaches can be implemented on proxy servers of departments, faculties and campus, to reduce the response time and save the network bandwidth of server. The proposed approaches do not consider multiple caching proxies, which cooperate and share their caches. In addition, regular retraining of classifiers is expected to improve the adaptability and efficiency of the proposed intelligent caching approaches. Eventually, instead of stand-alone web caching, intelligent Greedy-Dual approaches can be effectively integrated with a prefetching policy in order to improve the web performance.

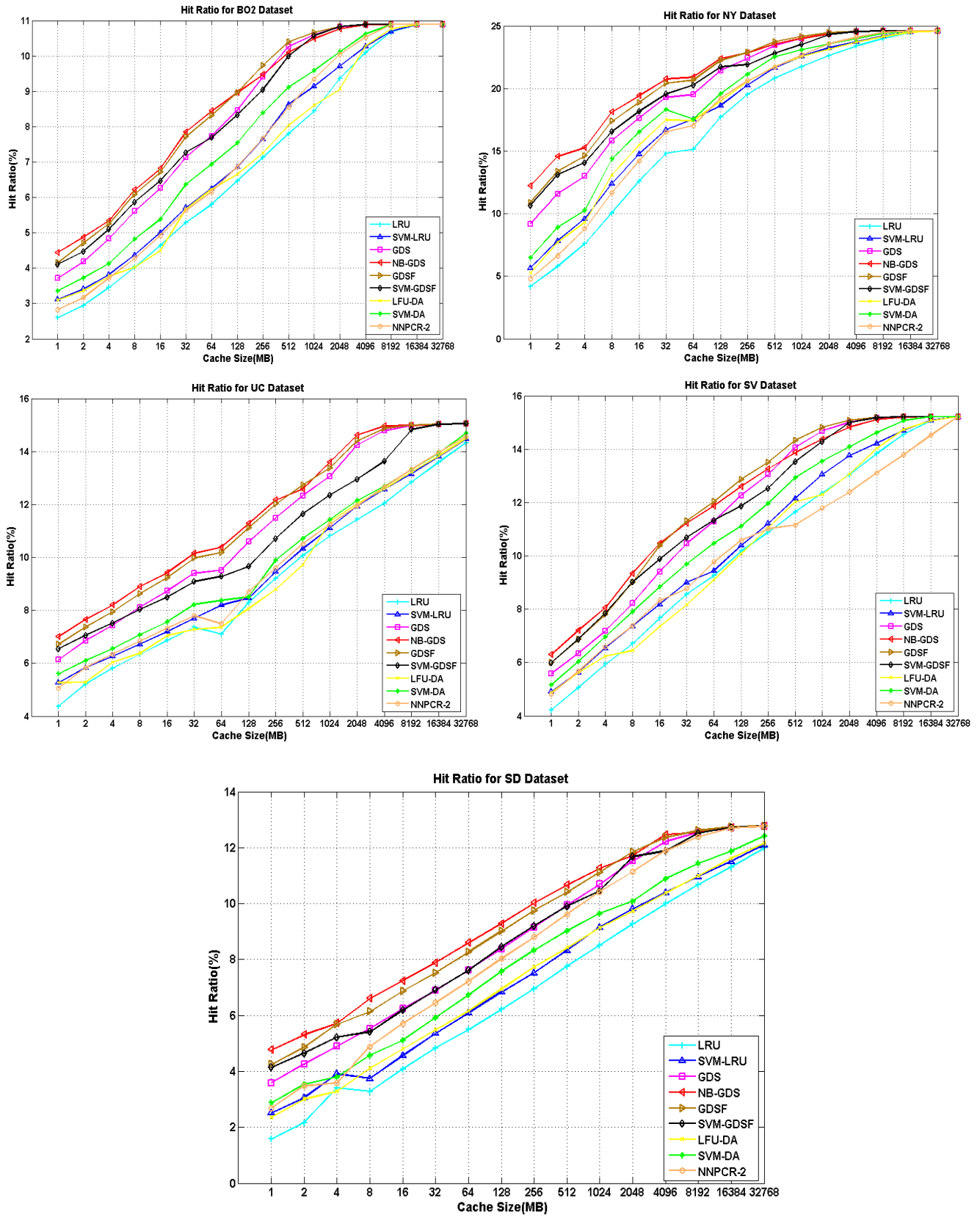


Fig. 2. Comparison of hit ratio between the conventional and intelligent web proxy caching approaches.

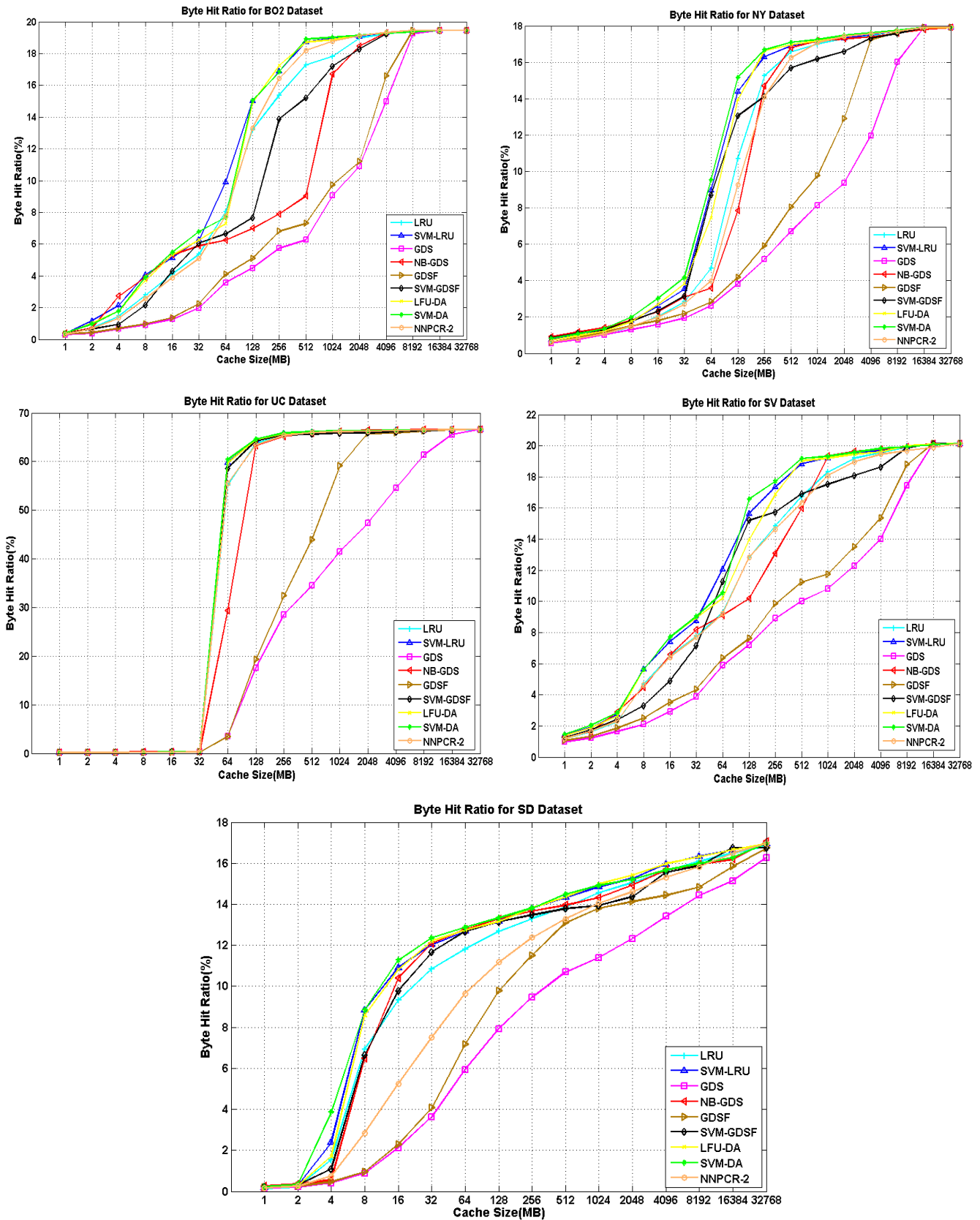


Fig. 3. Comparison of byte hit ratio between the conventional and intelligent web proxy caching.

ACKNOWLEDGMENT

This work was supported by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant No. (830-751-D1435). The authors, therefore, gratefully acknowledge the DSR technical and financial support.

REFERENCES

- [1] U. Acharjee, "Personalized and Artificial Intelligence Web Caching and Prefetching," MSc, University of Ottawa, Canada, 2006.
- [2] C. Kumar, and J. B Norris, "A new approach for a proxy-level web caching mechanism," *Decision Support Systems*, vol. 46(1), pp. 52-60, 2008.
- [3] C. Kumar, "Performance evaluation for implementations of a network of proxy caches," *Decision Support Systems*, vol. 46(2), pp. 492-500, 2009.
- [4] C. C. Kaya, G. Zhang, Y. Tan, and V. S. Mookerjee, "An admission-control technique for delay reduction in proxy caching," *Decision Support Systems*, vol. 46(2), pp. 594-603, (2009).
- [5] H. ElAarag, "A quantitative study of web cache replacement strategies using simulation," In *Web Proxy Cache Replacement Strategies*, Springer, London, pp. 17-60, 2013.
- [6] S. Romano, and H. ElAarag, "A neural network proxy cache replacement strategy and its implementation in the Squid proxy server," *Neural Computing & Applications*, vol. 20, pp. 59-78, 2011.
- [7] W. Ali, and S.M. Shamsuddin, "Intelligent Dynamic Aging Approaches in Web Proxy Cache Replacement," *Journal of Intelligent Learning Systems and Applications*, vol. 7(4), pp.117-127, 2015.
- [8] W. Ali, S. Sulaiman, and N. Ahmad, "Performance Improvement of Least-Recently-Used Policy in Web Proxy Cache Replacement Using Supervised Machine Learning," *International Journal of Advances in Soft Computing & Its Applications*, vol. 6(1), pp.1-38, 2014.
- [9] T. Koskela, J. Heikkonen, and K. Kaski, "Web cache optimization with nonlinear model using object features," *Computer Networks*, vol. 43, pp. 805-817, 2003.
- [10] T. Chen, "Obtaining the optimal cache document replacement policy for the caching system of an EC website," *European Journal of Operational Research*, vol. 181(2), pp. 828-841, 2007.
- [11] P. Cao, and S. Irani, "Cost-Aware WWW Proxy Caching Algorithms," In *Usenix symposium on internet technologies and systems*, Monterey, CA: pp. 193-206, 1997.
- [12] L. Cherkasova, "Improving WWW proxies performance with greedy-dual-size-frequency caching policy," *Hewlett-Packard Laboratories*, 1998.
- [13] R.C. Chen, and C.H. Hsieh, "Web page classification based on a support vector machine using a weighted vote schema," *Expert Systems with Applications*, vol. 31(2), pp. 427-435, 2006.
- [14] SH. Lu, DA. Chiang, HC. Keh, and HH. Huang, "Chinese text classification by the Naive Bayes Classifier and the associative classifier with multiple confidence threshold values," *Knowledge-Based Systems*, vol. 23(6), pp. 598-604, 2010.
- [15] Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Philip SY, and Zhou ZH, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14(1), pp.1-37, 2008.
- [16] CJ. Huang, YW. Wang, TH . Huang, CF. Lin, CY. Li, HM. Chen, PC. Chen, and JJ. Liao, "Applications of machine learning techniques to a sensor-network-based prosthesis training system," *Applied Soft Computing*, vol. 11(3), pp. 3229-3237, 2011.
- [17] JR. Quinlan, "C4.5: Programming for machine learning.. Morgan Kauffmann, 1993.
- [18] MC. Calzarossa, and G. Valli, "A fuzzy Algorithm for web caching," *SIMULATION SERIES*, vol. 35(4), pp. 630-636, 2003.
- [19] K. Tirdad , F. Pakzad, and A. Abhari, "Cache replacement solutions by evolutionary computing technique," In *Proceedings of the 2009 Spring Simulation Multi conference*, San Diego, California: International Society for Computer Simulation. p. 128, 2009.
- [20] A. Vakali, "Evolutionary techniques for Web caching," *Distributed and Parallel Databases*, vol. 11(1), pp. 93-116, 2002.
- [21] C. Yan, L. Zeng-Zhi, and W. Zhi-Wen, "A GA-based cache replacement policy," In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 2004.
- [22] W. Yasin, H. Ibrahim, NI. Udzir, and NA. Hamid, "Intelligent Cooperative Least Recently Used Web Caching Policy based on J48 Classifier," In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services 2014 Dec 4* (pp. 262-269), ACM, 2014.
- [23] S. Sulaiman, S.M. Shamsuddin, and A. Abraham, "Intelligent web caching using machine learning methods," *Neural Network World*, vol. 21(5), pp. 429-452, 2011.
- [24] J. Cobb, and H. ElAarag, "Web proxy cache replacement scheme based on back-propagation neural network," *Journal of Systems and Software*, vol. 81, pp. 1539-1558, 2008.
- [25] S. Sulaiman, SM. Shamsuddin, F. Forkan, and A. Abraham, "Intelligent Web caching using neurocomputing and particle swarm optimization algorithm," In *2008 Second Asia International Conference on Modelling and Simulation (AICMS)*; 2008 May 13; Kuala Lumpur: IEEE. pp. 642-647, 2008.
- [26] A. A W, and S.M Shamsuddin, "Neuro-fuzzy system in partitioned client-side Web cache," *Expert Systems with Applications*, vol. 38, pp. 14715-14725, 2011.
- [27] G. Sajeev, and M. Sebastian, "A novel content classification scheme for web caches," *Evolving Systems*, vol. 2, pp. 101-118, 2011.
- [28] A.P Foong, H. Yu-Hen, and D.M Heisey, "Logistic regression in an adaptive Web cache," *Internet Computing*, vol. 3, pp. 27-36, 1999.
- [29] National Lab of Applied Network Research (NLNR) (2010) Sanitized access logs: Data collected between 21st August and 4th September , 2010, <http://www.ircache.net/>.
- [30] N. Markatchev, and C. Williamson, "Webtraff: A GUI for web proxy cache workload modeling and analysis," In *10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*; IEEE. pp. 356-363, 2002.