Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015
ISSN 2074-8523

Intelligent Web Objects Prediction Approach in Web Proxy Cache Using Supervised Machine Learning and Feature Selection

Amira Abdalla[1], Sarina Sulaiman[1], Waleed Ali [2]

UTM Big Data Centre,
Ibnu Sina Institute for Scientific and Industrial Research,
Department of Computer Science, Faculty of Computing,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia,
e-mail: amiraalbana909@gmail.com; sarina@utm.my
[2]Faculty of Computing and Information Technology,
King Abdulaziz University, Saudi Arabia
e-mail: waleedalodini@gmail.com

Abstract

Web proxy cache is used to enhance the performance of network by keeping popular web objects in cache of proxy server for closer access. Intelligent approaches aim at improving the performance of conventional strategies. Mostly focus was on improving prediction mechanism, to guess the ideal objects that will be revisited in future; cache them and combine the result with the conventional algorithm. This research proposes an improved prediction method using automated method to select the influence features that produce accurate prediction results before combining with conventional algorithm. The method use supervised machine learning based on Naïve Bayes (NB) and Decision Tree (C4.5). It applies wrapper feature selection to specify influence features with optimal subset to improve the predictive power. Additionally two more features are extracted to know user's interest to make a smart and a wise decision for caching. The results showed that reduction for the number of features has a good impact on reducing computation time. Moreover, optimal subset selection achieves high performance and enhances accuracy.

Keywords: *Web Object Prediction, Web Proxy Cache, Wrapper Feature Selection, Feature Extraction, Supervised Machine Learning.*

1 Introduction

Web proxy cache is considered critically important in deployment of web caching system for improving the overall performance of the web system. This also solves problems affecting performance such as: user perceived delay, increase server load, and network traffic congestion. Focus of web proxy cache replacement is on determining worst cached objects and making decision to exchange it with new object instead in case when cache has no more space. Cache size being limited, require intelligent decisions for freeing the space and removing older objects or objects which may not be required in future. Web proxy cache replacement strategy is considered most effective solution to enhance content delivery to the users. It stores web requests that might be re-accessed in future. Though many cache replacement strategies exist each having different logic for selection of worst cached objects which need to be removed. The core objective of cache replacement strategies is to increase cache hit rate[1].

Mostly strategies try to solve problem of replacement[2].Conventional approaches are insufficient as their decision depends on one factor while ignoring most others. The important factors for cache replacement include size, recency, frequency, access latency, expiration time, modification time and fetching cost for an object. Conventional strategies lead to cache pollution problem[3, 4]; where frequently requested objects are kept in cache although they may not be revisited again. Newer intelligent strategies combine factors using conventional algorithms to improve the performance. Accurate prediction of object going to be revisited is still an open issue requiring more attention. The log file trekking behavior of users is another good idea; this can be utilized to provide full knowledge for training the predictor to predict re-visited objects with the help of interest of users. Some of intelligent schemes utilize different machine learning techniques for predicting revisited objects for example neural network, fuzzy logic, Support Vector Machine (SVM), Naïve Bayes (NB) and Decision Tree (C4.5)[4-8].

Features used during training phase influence the prediction process strongly. Existing intelligent techniques select features manually, hence the propose technique will employ an automatic method to precisely select the influencing features or the optimal subset of features that may provide the best prediction. For the purpose wrapper feature selection is applied with common machine learning classifiers as NB and C4.5 [4-9]. These classifiers have already proved their worth for providing enhanced prediction results in the field. The study shows that better feature selection not only improves prediction accuracy but also it reduces the computation time while using less features and reflect the important feature that effect on the prediction procedure. The paper is organized as follow: section 1 is introduction. Section 2 gives background and the related works. Section 3 covers theoretical background. Section 4 introduces the proposed method. Section 5 is about data collection and preprocessing. Section 6 reports evaluation and do comparative discussion. Section 7 concludes the paper.

2 Background

2.1 Web proxy cache replacement

Initial dating of Web caching pioneer works reaches to 1990; it rolled out as solution to the problems related to growth of World Wide Web (www). The problems are listed as: user perceive delay, congestion in network traffic and intensive server load condition. Main benefits of web caching is minimizing user perceived delay, reducing network traffic and reducing loads over servers. The method keeps copies of frequently used web objects on servers nearest to clients instead of directing all requests to main servers and responding from there. As web cache is used to improve performance of web system, so it can be deployed in different levels: client level, proxy level or original server level [10-12].

The focus of this study is at web proxy cache; since it's a common policy of managing web request over local or wide networks and servicing them simultaneously sometime staying behind proxy is pivotal between main server and the client side. Web cache strategy has three issues affecting its performance: cache replacement, cache consistency and cache placement. Web proxy cache replacement trek to decide which the worth cached object to remove in cache loaded conditions. Web proxy cache replacement focuses effective content delivery by storing web requests that might be re-accessed in near future more.

2.2 Related Work

Most of intelligent web proxy cache approaches uses cache replacement scheme to ascertain the objects may be required by user on basis of prediction classifiers referring to user interest. The Output of these classifiers is then integrated with conventional algorithm to effectively make the decisions more precise and intelligent. One objective is to mark most suitable object for replacement in situations when cache is filled and still user requests are waiting. Strength of prediction greatly is influenced by feature used for training. An automated method is developed to select the influence feature before training the classifier, since the selection of this features is more important to give accurate results before incorporate with traditional strategies to enhance the decision making. As we observed, all the previous intelligent approaches used manual methods for selecting the features that indicate the user interest before utilize in training the log file dataset. This manual method is insufficient. Therefore, the wrapper feature selection is applied to select the influence features that can accurately predict which object will be re-accessed again. Subsequent section mentions the related works in this field with focusing on the features used in the training phase.Table.1 reflect that related work.

Ali and Shamsuddin[5] provided an approach based on A neuro-fuzzy system (ANFIS) to predict the objects that could be revisited in future ,they selected manually the features of timestamp, frequency, delay time and size for training dataset. Romano and Elaarag[6] .Cobb and Elaarag [7] used neural network in prediction phase and the decision of replace the object has been specified depending on Back-Propagation Neural Network (BPNN), Neural Network Proxy Cache Replacement (NNPCR)[6] and NNPCR-2[1],the features used for training dataset are size, frequency and recency. Mohmed [13] used different features including number of hits, response time, script size, CPU usage and bandwidth, to integrate the result of BPNN for decision and least recently used (LRU) algorithm for replacement, Features inside training dataset are: i) size of script, ii) bandwidth, iii) total hits, iv) processing required. Sulaiman et al.[14] came up with these features for training dataset i) total hits, ii) size of script and iv) time taken. The research proposed particle swarm optimization (PSO) on top of Mohamed's strategy by to improve neural network performance. Multilayer perceptron network (MLP) classifier was used in web proxy caching by Koskela et al.[15].

MLP predicts the class of web object, and for training the dataset, select features like i) server responses in form of HTTP and ii) syntactic features from HTML document. Foong et al.[16] utilized a logistic regression model (LR) for prediction phase. The features used for training dataset are i) object type, ii) object recency, iii) object frequency with backlogged history and iv) object's size. While Tian et al.[17] proposed an intelligent predictor mainly based upon BPNN instead of the LR model. Features of Training dataset are i) object size, ii) frequency, iii) time of retrieval, iv) object type and v) relative frequency. For classification of cached objects to multilevel classes Sajeev and Sebastian[18] employed Multinomial Logistic Regression (MLR). The features of training dataset used are i) delay, ii) object popularity, iii) object size, iv) recency, v) type of object and vi) consistency of popularity.

For better prediction of re-accessible objects Benadit et al.[19] used semi supervised learning mechanism called Expectation Maximization Naive Bayes classifier (EM-NB). The improved the performance by incorporating result with LRU and GDSF. Feature used are i) recency , ii) frequency, iii) size, iv) Sliding Window Length frequently (SLW-frequently) and iv) number of estimated future request. For prediction Ali et al.[20] used Adaptive Neuro-Fuzzy Inference System (ANFIS) . Results were combined to improve the performance of LRU. The features used are i) time stamp, ii) frequency, iii) delay time and iv) object size. Supervised machine learning Naïve Bayes (NB) ,Support Vector Machine (SVM) and decision tree (C4.5) are used by Ali et al.[3,4,9] for better prediction Results are combined LRU, GDS and Greedy Dual Size Frequently (GDSF) to enhance the performance. Features used: i) recency, ii) elapsed time, iii) frequency, iv) SWL- frequently, v) object size and vi) type of object.

Table 1: The previous related works that selected their features for Training phase manually

Author& date	Title	Data used	Feature used for training	Classifier used in prediction phase
Benadit et al.[19] -2015	Expectation Maximization Naive Bayes classifier (EM-NB).	The data of one day.provide by The five proxy logs files(BO2-NY-SD-UC-SV) of IRCache (2010-NLANR).	Recency ,frequency ,size,swl-frequency and number of future requests.	Expectation maximization Naive Bayes classifier.
Ali et al.[3] -2014	Performance Improvement of Least-Recently-Used Policy in Web Proxy Cache Replacement Using Supervised Machine Learning.	The data of one day.provide by The five proxy logs files(BO2-NY-SD-UC-SV) of IRCache (2010-NLANR).	Recency,retrieval time, size, frequency, swl-frequncy and object type.	SVM-NB-C4.5
G.Sajeev and M. Sebastian [21] -2013	Building semi-intelligent web cache systems with lightweight machine learning techniques.	Data of proxy logs file (BO2) of IRCache (NLANR).	Independent variable,primary variables, popularity, recency, object size, auxiliary variables, popularity consistency, delay, type of object and dependent variable.	multinomial web object classifier
Ali et al[4] -2012	Intelligent Web proxy caching approaches based on machine learning techniques.	The data of one day .provide by The five proxy logs files(BO2-NY-SD-UC-SV) of IRCache (2010-NLANR).	Recency,retrieval time,size,frequency,swl-frequncy and object type.	SVM-C4.5
Ali et al.[9]	Intelligent	The data of	Recency,retrieval	NB Classifier

-2012	Naive Bayes-based approaches for web proxy caching.	one day .provide by The five proxy logs files(BO2-NY-SD-UC-SV) of IRCache (2010-NLANR).	time,size,frequency,swl -frequncy and object type.	
Romano and Elaarag[6] -2011	A neural network proxy cache replacement strategy and its implementation in the Squid proxy server.	The proxy logs files of IRCache -2005-NLANR.	Size, frequency and recency.	Back-Propagation Neural Network (BPNN).
Sajeev and Sebastian[18] -2011	A novel content classification scheme for web caches	The data of one day from the IRCache network (proxy logs files-2010).	delay,object popularity, object size, recency,type of object and consistency of popularity.	Multinomial Logistic Regression (MLR)
Ali and Shamsuddin[5] - 2009	Intelligent client-side web caching scheme based on least recently used algorithm and neuro-fuzzy system.	BU Web traces,1995 Offer By Cunha of Boston University.	timestamp, frequency, delay time and size.	A neuro-fuzzy system (ANFIS).
Cobb and Elaarag [7] -2008	Web proxy cache replacement scheme based on back-propagation neural network.	The proxy logs files of IRCache -2005-NLANR.	size, frequency and recency	(BPNN).
Sulaiman et al.[14] -2008	Intelligent Web caching using neurocomputing and particle swarm optimization algorithm	1995-BU-Web trace.	total hits, size of script and time taken	Using pso for enhance mohamed's approach
Mohmed [13]- *2006*	Intelligent Web caching architecture	from entity Sdn.Bhd company,	size of script, bandwidth, total hits, processing required	Integrated BPNN and LRU

		Malaysia		
Koskela et al.[15] -2003	Web cache optimization with nonlinear model using object features	Provide by Finnish University and Research Network web cache (FUNET).	server responses in form of HTTP and,syntactic features from HTML document	Multilayer perceptron network (MLP) classifier was used
Tian et al.[17] -2002	An adaptive web cache access predictor using neural network	BU Web Trace in 1995.	object size, frequency, time of retrieval,object type and relative frequency	based upon BPNN .
Foong et al.[16] -1999	Adaptive Web caching using logistic regression	five server logs file obtainable at Internet Traffic Archive.	object type, ii) object recency, iii) object frequency with backlogged history and iv) object's size.	A logistic regression model (LR) for prediction phase.

2.3 Feature Selection

Feature selection is an important part of machine learning procedure. Selection of discriminating features influences the predictive power and accuracy of the classifier using them. However, utilizing all available features is quit impractical since the quantity of the available training data is usually low with respect to dimensionality[22]. Feature selection not only highlights the importance of features[23], establishes a trade-off between the adequacy of the learned model

and the number of selected features. Selecting a set of influence features to enhance the predictive power of a classifier is a difficult task. The feature selection process aims at decreasing the complexity and improving the accuracy of classification. In addition, feature selection can reduce computation time since, it chooses the best subset and reduces the number of feature uses this advantages was a motivation to apply it in our big dataset. The log file contains a big data, so when we select the best subset it reduces the time of training data. There exist multiple ways to evaluate the features. These methods of evaluation had been categorized into three classes a). wrapper based evaluation, b) filter based evaluation and c) hybrid evaluation[24].

Filter base evaluation extract feature and assign evaluation weight but no data classification is done. After few repetitions eventually it comes up with ‘good’ feature subset. Features extraction criterions mostly in this class are statistical by design. Filter base evaluation techniques extract features subset on basis of high dependency on target class and less inter correlation. This class uses statistical measures to evaluate and weigh the features while maximizing the cluster performance. Inversely, the wrapper-based approaches utilize inductive algorithms for classification to measure the goodness of features subset.[25, 26] Few researches mention wrapper based evaluations methods can perform better than filter based techniques. The core objective of feature selection is to eliminate features which contain noise, or are redundant in some way, or are irrelevant features from raw data sets while keeping the loss of information minimal. In large data sets extracting a minimum feature subset is a challenging task.[26] However, Feature selection mainly depends on two components:

a) Evaluation strategies include wrapper and, filter methods. The research focuses on wrapper method because it is compatible to machine learning [22] and considered better than filter method.

b) Search engine include optimum, heuristic and randomize strategies.

2.3.1 Wrapper Methods of Feature Selection

The wrapper method does feature subset selection and applies the induction algorithm as a black box[27]. Black box uses the interface without knowledge of algorithm. The induction algorithm is considered the part of the evaluation function in feature subset selection algorithm which is used to search the optimal feature subset. Fig.1 shows the working of wrapper feature selection approach[27]. This approach can assess the goodness of all the selected feature subset utilizing the induction algorithm over original dataset. Wrapper approach has the ability to specify potential feature subsets accurately. The feature subsets also correspond with learning algorithms [27-30]. The wrapper method performs search in space of all possible parameters. The wrapper search has some pre requisites like a termination condition, a state space, and a search engine which is investigated in subsequent sections. This study uses Wrapper SubsetEval as an attribute evaluator using Weka 3.7.12 and the best first as search method.

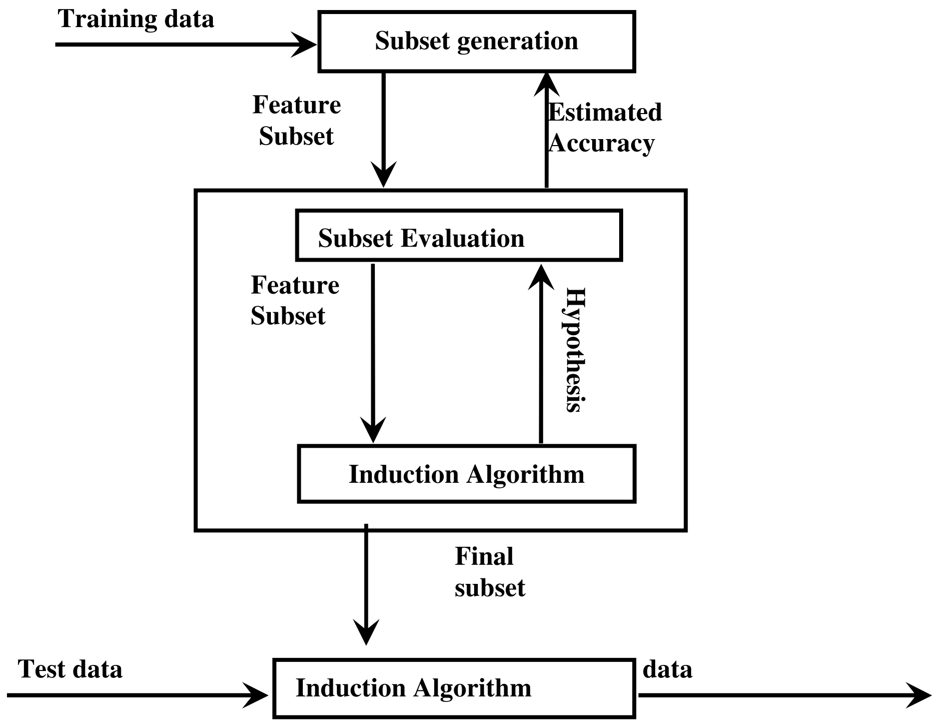

Fig. 1: The Wrapper Approach for Feature Selection [27]

Wrapper SubsetEval:

SusetEval evaluates attribute sets with the help of learning scheme. It selects an optimal subset for producing the best result. Cross validation is done to estimate the accuracy of the learning scheme for a set of attributes. Many search engines are packed with wrapper like best- first and hill-climbing search. However the research uses best- first method.

The best first searching engine (BFS):

The BFS is a Linear Forward Selection (LFS) search engine. A limited number of attributes (k) are taken into consideration by LFS. The method either select the top k attributes by initial ordering or carry put a ranking. It searches the space of attribute subsets by greedy hill climbing augmented with a back tracking facility[27]. Setting the number of consecutive non-improving nodes allow the level control of backtracking. Best first can start with an empty set of attributes and perform a forward search, or it can start with the full set of attributes and search backward, or start at any point then search in both directions (by considering all possible Single attribute additions and deletions at a given point). Some used types of selection algorithms:

1. Forward selection (FS): this strategy is considered easiest as compared to other wrapper selection algorithms[27]. This method starts the procedure

without having any feature in the feature subset and do like greedy strategy by adding features sequentially until no possible single feature addition results in a higher valuation of the induction function.

2. Backward Elimination (BE): Starts the procedure by considering all the features sets and sequentially deleting features as long as the valuation does not degrade.
3. Bi-directional: Begins its procedure at any point in space and search in both directions (by considering all possible single attribute for adding or removing at a given point)[30].

3 Theoretical Background

3.1 Naïve Bayes

The Naïve Bayes (NB) classifier is one of basic machine learning algorithm. The outcome **p(x|y)** of a probabilistic NB is said to be the probability of a pattern lying in class x found in result of observing data **y** (called posterior probability)[31, 32]. The assumption is that text data is result of set of parametric models (associating each to at-least one class). To estimate unknown parameters of the model training data is used. The Bayesian network forms an acyclic directed graph representing the probability distribution. Node in the directed graph is a symbol for event or a random variable and the arcs through the nodes stand for causal relationship connecting them. Through the operational stage, the classifier computes any model. The probability **p (x|y)** gives the probability of the model generated document. The Bayes theorem performs inversion of generative model and calculates posterior probabilities[33](probabilities generated for pattern). It performs final classification with selection of model giving max posterior probability. Though it's simple yet NB classifier is as precise as are learning algorithms for text categorization. Web application use NB classifier for focus crawling and recommending systems[32] mostly.

3.2 C4.5 Training

Quinlan[34] formulated C4.5 decision tree, which is most commonly practiced technique for classification purposes by applications in marketing, finance, medicine and engineering[35]. C4.5 utilizes inductive inference rule, organizes its learned classifier in set of if-then rules to match human understanding[36]. Consequently, e decision tree is uncomplicated to be clear for understanding. Furthermore, C4.5 has ability to produce good outcome when dealing with large dataset with swift, in addition to its ability to deal with nominal data[37].The C4.5 structures its training by built the decision tree in a top-down recursive representation. Then its learning implemented using the following steps, 1). Training patterns are placed at root, 2). Training patterns are split recursively using features extracted with help of impurity function which uses information gain ratio. 3) repeat splitting process unless all patterns are placed at their specific class. At the end probabilities of classes are computed using relative frequency of each class at leaf[34].

4 Proposed Method

4.1 Intelligent Web Objects Prediction Approach in Web Proxy Caching using Supervised Machine Learning and Wrapper Feature Selection

After we completed data pr-eprocess phase as we described in section 5, we obtained our data contain ten features including URL, time stamp, total elapsed time, object size, frequently ,swl-frequently, recency, the type of Web object ,time spent and mean as shown in Table 2.

Then we apply feature selection using wrapper method before training phase to choose automated the best subset that produce the best result with each classifier .

Table 2: All features used before using Wrapper feature selection

No	The feature name	Description	How to calculate
1	Time stamp	It is the sequential time of requests in millisecond.	Given.
2	Elapsed time	Retrieval time of Web object.	Given.
3	frequently	Frequencies of Web object.	The numbers of frequent occur of web object.
4	Swl-frequently	Object's frequency based on backward-looking sliding window.	$swl-freq=\begin{cases} swl-freq_{i-1}+1, if\ \Delta t \le swl \\ 1 \quad \text{otherwise} \end{cases}$
5	Size	Web objects Size.	Given.
6	Type class	Web objects type.	Given.

7	Recency	Recency of Web object based on backward-looking sliding window.	$recency = \begin{cases} \max(swl, \Delta t) \text{ if object was requested before} \\ swl \text{ otherwise} \end{cases}$
8	url-id	Identifier number of url to convert to integer number.	Randomly generated identifier number for each url
9	Time spend	The time that spent in each page in millisecond.	The different timestamp in millisecond between current page and next one which immediately followed.
10	Mean	The average of time spent in each page in millisecond.	The sum of time spent in each page for all its frequents divide by its number of frequents.

Table 3: The selected best subset for each classifier in different datasets

The best subset	**BO2**	**NY**	**UC**	**SV**	**The a average of accuracy**
NB	elapsed ,url-id	timestamp, url_id, recency, typeclass	time stamp, size , url_id, recency, freq, swl_freq, timespend.	timestamp, url_id, recency	95.22
the number of features	2	4	7	3	
C4.5	timestamp, url id,swl frequncy	timestamp, url_id and freq	time stamp, elapsed, size ,recency, freq ,swl_freq, mean, typeclass.	timestamp, elapsed, size ,url_id, recency, freq, typeclass.	95.77
The number of features	3	3	8	7	

4.1.1 Training Phase

After completing the first phase of preprocessing and the second phase of feature selection, each classifier is shown in Table 3 along with best feature subset selected for each dataset. The third phase is the training phase in which subset of feature are utilized which are chosen by applying feature selection. Weka 3.7.12 (Waikato Environment for Knowledge Analysis) is sued with Delphi language, wrapper as attribute evaluator and best first as a search method, then best selected subset of features is used to produce specific machine learning algorithm for training process. The machine learning algorithm is trained and evaluated based upon the dataset and selected features. Decision tree (J48) is used as a classifier in training phase and apply cross validation using 10 folds, J48 learning algorithm is a Java coded version of C4.5 and is compatible with WEKA. After training the decision tree, a test instance of any web object can be classified depending on traversing the tree in a top-down way based upon test results of instance unless it reaches a leaf node. The leaf node is representation of predicted class, determining the re-visit probability of an object. The second classifier that we used is the Naïve Bayes (NB), this algorithm is prepared for training, by discretizing the proxy datasets employing MDL as suggested by Fayyad and Irani [38] in most recommended setup in WEKA similar to[9].

5 Data Collection and Pre-processing

The data of the proxy logs files and traces had been gained from four proxy servers of IRCache network of US. The data is spanned over a time of fifteen days (NLANR, 2010a). The datasets used are BO2, NY, SV, UC gathered in time period from 21-8-2010 till date 4-9-2010[4]. This study uses the proxy logs files of 21-08-2010 for the training phase. The trace has significant influence on the performance. So, to obtain result that reflects the correct classification we must make a correct trace preparation. The commonly features inside an access proxy log entry are: i) time elapsed, ii) timestamp, iii) object size, iv) client address, v) log tag with HTTP script, vi) URL, vii) identification of user, viii) request method, ix) content type, x) hierarchy of data and the hostname.

Firstly two sub-features are extracted from timestamp feature, because a feature is required that can reflect user interest for improving the prediction power since, The process is based on user interest, so the first sub-feature extracted is Time Spent. It shows the time user spent at each page after it was requested. The time spent value is one measure of user interest. If a user spends short time at some page after request, it shows less interest. On contrary spending long time at some page after it was requested infers more interest. This also tells that this page be revisited again in future. Time spent is computed for each current request as the difference between current request and the subsequent request which may come after the current.Second feature to compute is Mean-time or Mean (in short) which is average of all the time spent at that page. Mean is calculated by computing sum of Time Spends at each page (request), and then dividing by the number of its frequency(number of visited times). After features are extracted the data pre-processing step similar to[4] is done. Data pre-processing filters irrelevant and invalid requests which are logged at proxy server. The steps to do this are following:

- **Parsing**: is the process of detecting border lines between records in sequence in log files, also finding the unique fields inside each record.
- **Filtering**: is the process of removing irrelevant records for example requests which can't be cached (queries containing "?" inside URL and cgi-bin) and records having failed HTTP status codes. The only records are considered who have success status having code 200.
- **Finalizing**: is a process of eliminating not required fields. Additionally each distinct URL is transformed into integer variable to decrease the simulation time. Finally, our data contain ten features including URL, time stamp, total elapsed time, object size, frequently, swl-frequently, recency, the type of Web object, time spent and mean as shown in Table 1.

6 Evaluation and Comparison Discussion

As we mentioned in section 5 the dataset that we used is a data of a log file and it's a big data so we take a sample for training phase show table 4.

Table 4: Description of the Proxy datasets used in this study

	BO2	NY	UC	SV
Proxy location	Boulder, Colorado	New York, NY	Urbana-Champaign, Illinois	Silicon Valley, California (FIX-West)
Duration of collection	21/8-4/9/2010	21/8-4/9/2010	21/8-4/9/2010	21/8-4/9/2010
Total request	1,210,693	3,248,452	8,891,764	2,496,001
Training patterns	19,525	40,046	91,648	38,829

Two measures are used in this study to evaluate the performance of a classifier i) correct classification rate (CCR) and evaluate the computation time.

$$CCR = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where True Positive (TP) means the number of correct classifications of the positive examples .True Negative (TN) means the number of correct classifications of negative examples. False Positive (FP) means the number of incorrect classifications of negative examples .False Negative (FN) means the number of incorrect classifications of positive examples.

The important class considered positive is class of objects to be re-accessed again (using forward SWL, take 1 in target class). The objects belong to class positive (one), otherwise it is classified as negative class member. As mentioned before the positive class is the minority and the most object revisited is once or few time only. Table 5 gives the measure that is utilized to weight the classifiers for machine learning. This measure is utilized to evaluate and compare with previous studies in same field, C4.5 that using wrapper feature selection (WFS) and NB (WFS) employing diverse datasets (four in totals).

The testing results are compared with the results of cross validation used 10 folds. Highest values were bold and the lowest values were underlined. It is also noticed that higher CCR is achieved using all datasets (signed as bold) which is greater than BPNN, C4.5 and NB who choose their features manually. This proves that feature selection enhanced the performance of machine learning classifiers in addition to reduction of number of features required (refer to Table 2). In testing ,we observed the average of CCR for C4.5(WFS), NB(WFS), BPNN, C4.5[4] and NB[9] are 95.77 % , 95.22 % , 87.422% , 94.55% and 94.428 % , respectively.

Table 5 : Compare the CCR measure among different datasets

CCR	**BO2**	**NY**	**UC**	**SV**	**AVG**
NB (WFS)	**96.05**	**94.07**	**96.51**	**94.27**	**95.22**
C4.5(WFS)	**96.38**	**95.44**	**96.73**	**94.56**	**95.77**
BPNN	92.21	80.41	90.15	88.06	87.788
NB	95.57	91.79	95.71	93.86	94.428
C4.5	95.6	92.19	96.03	94.08	94.55

Furthermore, computation time for classifiers is compared show table 6; NB is faster classifier then C4.5 but BPNN is slower as it takes a long time for the training of data. It is observed that C4.5 algorithm takes less time for all dataset except for UC and SV. One reason is that both datasets is larger than others in size, so select more numbers of features in its best subset which increases the computation time of training. Although that the average computation time is less and better for NB and then C4.5 which is 0.01 ,1.46, respectively when compared to NB[9] and C4.5which is 0.5975 and 1.315, correspondingly.

Table 6 : Compares the computation time among different datasets

TIME	**BO2**	**NY**	**UC**	**SV**	**AVG**
NB (WFS)	**0**	**0.01**	**0.02**	**0.01**	**0.01**
C4.5(WFS)	**0.22**	**0.48**	3.92	1.22	1.46

BPNN	146.59	316.34	701.18	303.40	693.36
NB	0.12	0.35	1.59	0.33	0.5975
C4.5	0.36	0.85	**3.36**	**0.69**	**1.315**

clearly shows that the NB and C4.5 achieves good results but due to imbalance dataset the standard classifier ignores the minority class or remains undiscovered; consequently, misclassified test sample of minority class occurs more often than that of majority class[39]. Class with highest patterns is predicted the majority class. As noticed the machine learning algorithms NB and C4.5 work better than neural network in spite of the imbalance in the dataset. This confirms that, the NB and C4.5 are capable of predicting minority class consisting objects that potentially be re-visited later on.

7 Conclusion and Future work

In this paper, an improvement in the prediction power of intelligent web caching is presented by applying wrapper feature selection. Additionally two new extracted features are proposed for increased determination of user's interest. Feature selection is quite significant for classification process. The scheme can extract most influenced feature that can improve the results of classifier for best performance. The proposed scheme can reduce the number of feature and can reduce dimensionality of dataset. Also one contribution is the reduction in computation time by selecting the optimal subset. The intelligent web caching is enhanced by predicting accurately the re-accessed objects in near future. This is then integrated with conventional methods to enhance them as well. The experiments executed shows that accuracy and computational time is improved with selection of influence features. Moreover, apply wrapper method helped to attain accurate and better prediction. The results show the NB and C4.5, using automatic method with Wrapper Feature selection achieves high average of accuracy that is **95.22%** and **95.77%** respectively. Compared with existing schemes works which are manual in nature NB and C4.5 achieve **94.43%** and **94.55%,** correspondingly. As we notice the datasets that we used were data of log file so, considered as a big data since its quantity is big and varies in data types and it changes daily(velocity).

In future work, we will broaden our scope to deal with more automated methods and use a new technique of big data to pre-process and analysis to improve predictive power to achieve better result.

AKNOWLEDGEMENTS

The research work is sponsored by The Ministry of Higher Education, Malaysia and collaboration with Research Management Centre (RMC), Universiti Teknologi Malaysia. This research is financial supported by UTM Flagship Grant (Q.J130000.2428.02G70), FRGS Grant (R.J130000.7828.4F634) and E-Science Fund (R.J130000.7928.4S117). Smart Digital Community Research Alliance, UTM and Soft Computing Research Group (SCRG) is acknowledged for the inspiration in making this study a success. We pay sincere thanks to other researchers who paved the way for this work.

References

[1] S. Jarukasemratana and T. Murata, "Web caching replacement algorithm based on web usage data," *New Generation Computing,* vol. 31, pp. 311-329, 2013.

[2] S. Podlipnig and L. Böszörmenyi, "A survey of web cache replacement strategies," *ACM Computing Surveys (CSUR),* vol. 35, pp. 374-398, 2003.

[3] W. Ali*, et al.*, "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, 2014.

[4] W. Ali*, et al.*, "Intelligent Web proxy caching approaches based on machine learning techniques," *Decision Support Systems,* vol. 53, pp. 565-579, 2012.

[5] W. Ali and S. M. Shamsuddin, "Intelligent client-side web caching scheme based on least recently used algorithm and neuro-fuzzy system," in *Advances in Neural Networks–ISNN 2009*, ed: Springer, 2009, pp. 70-79.

[6] S. Romano and H. ElAarag, "A neural network proxy cache replacement strategy and its implementation in the Squid proxy server," *Neural computing and Applications,* vol. 20, pp. 59-78, 2011.

[7] 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.

[8] H. ElAarag, "Implementation of a Neural Network Proxy Cache Replacement Strategy in the Squid Proxy Server," in *Web Proxy Cache Replacement Strategies*, ed: Springer, 2013, pp. 83-103.

[9] W. Ali*, et al.*, "Intelligent Naive Bayes-based approaches for web proxy caching," *Knowledge-Based Systems,* vol. 31, pp. 162-175, 2012.

[10]C. C. Kaya*, et al.*, "An admission-control technique for delay reduction in proxy caching," *Decision Support Systems,* vol. 46, pp. 594-603, 2009.

[11]C. Kumar, "Performance evaluation for implementations of a network of proxy caches," *Decision Support Systems,* vol. 46, pp. 492-500, 2009.

[12]C. Kumar and J. B. Norris, "A new approach for a proxy-level web caching mechanism," *Decision Support Systems,* vol. 46, pp. 52-60, 2008.

[13]F. Mohamed, "Intelligent Web caching architecture," Universiti Teknologi Malaysia, Faculty of Computer Science and Information System, 2006.

[14]S. Sulaiman, *et al.*, "Intelligent Web caching using neurocomputing and particle swarm optimization algorithm," in *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, 2008, pp. 642-647.

[15]T. Koskela, *et al.*, "Web cache optimization with nonlinear model using object features," *Computer Networks,* vol. 43, pp. 805-817, 2003.

[16]A. P. Foong, *et al.*, "Adaptive Web caching using logistic regression," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, 1999, pp. 515-524.

[17]W. Tian, *et al.*, "An adaptive web cache access predictor using neural network," in *Developments in Applied Artificial Intelligence*, ed: Springer, 2002, pp. 450-459.

[18]G. Sajeev and M. Sebastian, "A novel content classification scheme for web caches," *Evolving Systems,* vol. 2, pp. 101-118, 2011.

[19]P. J. Benadit, *et al.*, "Improving the Performance of a Proxy Cache Using Expectation Maximization with Naive Bayes Classifier," in *Computational Intelligence in Data Mining-Volume 2*, ed: Springer, 2015, pp. 355-368.

[20]W. Ali, *et al.*, "Patterns analysis and classification for web proxy cache," in *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*, 2011, pp. 97-102.

[21]G. Sajeev and M. Sebastian, "Building semi-intelligent web cache systems with lightweight machine learning techniques," *Computers & Electrical Engineering,* vol. 39, pp. 1174-1191, 2013.

[22]H. Liu, *et al.*, "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns," *Genome informatics,* vol. 13, pp. 51-60, 2002.

[23]Z. S. J. Hoare, "Feature selection and classification of non-traditional data: examples from veterinary medicine," University of Wales, 2007.

[24]I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[25]J. Novaković, *et al.*, "Toward optimal feature selection using ranking methods and classification algorithms," *Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043,* vol. 21, 2011.

[26]A. G. Karegowda, *et al.*, "Feature subset selection problem using wrapper approach in supervised learning," *International journal of Computer applications,* vol. 1, pp. 13-17, 2010.

[27]R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence,* vol. 97, pp. 273-324, 1997.

[28]N. Nnamoko*, et al.*, "Evaluation of Filter and Wrapper Methods for Feature Selection in Supervised Machine Learning," *Age,* vol. 21, pp. 33.2-11.8, 2014.

[29]P. Bermejo*, et al.*, "Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking," *Knowledge-Based Systems,* vol. 25, pp. 35-44, 2012.

[30]J. Leng*, et al.*, "A wrapper-based feature selection for analysis of large data sets," 2010.

[31]D. Heckerman, *A tutorial on learning with Bayesian networks*: Springer, 1998.

[32]D. Heckerman, "A tutorial on learning with Bayesian networks," in *Innovations in Bayesian Networks*, ed: Springer, 2008, pp. 33-82.

[33]N. Friedman*, et al.*, "Bayesian network classifiers," *Machine learning,* vol. 29, pp. 131-163, 1997.

[34]J. R. Quinlan, *C4. 5: programs for machine learning*: Elsevier, 2014.

[35]P. Chiu and G. Webb, "Using C4. 5 as an induction engine for agent modelling: An experiment of optimisation," in *Proceedings of the User Modelling conference UM*, 1997.

[36]G. I. Webb*, et al.*, "Comparative evaluation of alternative induction engines for Feature Based Modelling," *International journal of artificial intelligence in education,* vol. 8, pp. 97-115, 1997.

[37]C.-J. Huang*, et al.*, "Applications of machine learning techniques to a sensor-network-based prosthesis training system," *Applied Soft Computing,* vol. 11, pp. 3229-3237, 2011.

[38]K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," 1993.

[39]Y. Sun*, et al.*, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 23, pp. 687-719, 2009.