Meaningless to Meaningful Web Log Data for Generation of Web Pre-caching Decision Rules Using Rough Set

Sarina Sulaiman, Siti Mariyam Shamsuddin, Nor Bahiah Ahmad
Soft Computing Research Group
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, Johor, Malaysia
sarina@utm.my, mariyam@utm.my, bahiah@utm.my

Ajith Abraham
Machine Intelligence Research Labs (MIR Labs)
Washington, 98092, USA
http://www.mirlabs.org
ajith.abraham@ieee.org

Abstract—**Web caching and pre-fetching are vital technologies that can increase the speed of Web loading processes. Since speed and memory are crucial aspects in enhancing the performance of mobile applications and websites, a better technique for Web loading process should be investigated. The weaknesses of the conventional Web caching policy include meaningless information and uncertainty of knowledge representation in Web logs data from the proxy cache to mobile-client. The organisation and learning task of the knowledge-processing for Web logs data require explicit representation to deal with uncertainties. This is due to the exponential growth of rules for finding a suitable knowledge representation from the proxy cache to the mobile-client. Consequently, Rough Set is chosen in this research to generate Web pre-caching decision rules to ensure the meaningless Web log data can be changed to meaningful information.**

Keywords-component; decision rules; rough set; web caching; web pre-fetching; web log data

I. Introduction

Either hidden or meaningless information from Web log mining can be generated through significant rules [1,2,3]. Consequently, a proxy cache will record a Web logging of all clients that use the same server. However, a decision on location for architecture of Web pre-fetching (WP) engine will affect a prediction of Web objects [1]. In this case, to reduce usage of a mobile phone memory and to reduce the latency, the best solution is to put this engine near to the client-side, which means in the WP between Web clients and proxy cache [4,5].

Simultaneously, the generated rules from hidden Web log are essential to predict the accurate website from the proxy cache [1,6]. The precise rules for client query result are crucial to ensure that the predicted websites are correct.

Therefore, the aim of this research is to improve the existing schemes that consist of key algorithms serving for mobile clients including a cache replacement algorithm. The idea is to determine, which data items should be cached and evicted when free cache space are lacking. Moreover, a cache validation algorithm is responsible in maintaining data consistency between the mobile-client and Web cache server-side. Subsequently, a pre-caching algorithm will determine data items that should be cached in advance based on the generated rules from a proxy cache for Web objects prediction.

In this research, Web caching (WC) can be classified through a log dataset that conceals interesting behaviour and hidden information. The Web cache content used in this research is a log data from E-Learning@UTM (EL) Web server that had been monitored for two days and Boston University (BU) client server for seven months [7,8].

Furthermore, in Web cache content, the data needs to analyse and filter to identify either to cache or not to cache of Web contents from a cache server [9,10]. This data contains dissimilar parameters consist of Uniform Resource Locator (URL), size, retrieval time and others for Web cache contents [11,12]. The details process on pre-processing and normalise BU and EL data can be referred in [7,8].

This paper is organised as follows: Section 2 combines the related works of previous research on Rough Set (RS). Section 3 presents the experimental setup for this research. Sections 4 and 5 describe reduct, rules' generation and decision rules base on prediction, respectively. Next, Section 6 is about rules' derivation and classification results. Finally, a discussion, conclusions and future work are presented in Section 7 and 8.

II. Related Works

RS is as an approach to represent uncertainty of dataset. It is based on equivalence relations and set approximations, and the algorithms for computing RS properties are combinatorial in nature. The main advantages of Rough Set Theory (RST) are as follows [13]:

This research is supported by The Ministry of Higher Education (MOHE) and collaboration with Research Management Centre (RMC), Universiti Teknologi Malaysia. This paper is financial supported by University Grant (GUP) Program, Tier 1 (vote: Q.J130000.2528.03H00) and FRGS Grant (vote: R.J130000.7828.4F026).

(i) It does not need any preliminary or additional information about data;
(ii) It is easy to handle mathematically;
(iii) Its algorithms are relatively simple.

Moreover, Wakaki *et al.* [14] used the combination of the RS-aided feature selection method and the support vector machine with the linear kernel in classifying Web pages into multiple categories. The proposed method gave acceptable accuracy and high dimensionality reduction without prior searching of better feature selection. Liang *et al.* [15] used RS and RS based inductive learning to assist students and instructors with WebCT learning. Decision rules were obtained using RS based inductive learning to give the reasons for the student failure. Consequently, RS based WebCT Learning improves the state-of-the-art of Web learning by providing virtual student or teacher feedback and making the WebCT system much more powerful. These works focuses on RS to enhance classification accuracy for feature selection and decision rules. However, the question is how to ensure the relationship among the attributes in specific datasets?

In this case, to answer that question, Ngo and Nguyen [16] proposed an approach to search for results clustering based on tolerance RS model following the work on document clustering. The application of tolerance RS model in document clustering was proposed as a way to enrich document and cluster representation to increase clustering performance. Furthermore, Chimphlee *et al.* [17] presented a RS clustering to cluster Web transactions from Web access logs and used Markov model for next access prediction. Users can effectively mine Web log records to discover and predict access patterns while using this approach. Chimphlee *et al.* [17] performed experiments using real Web trace logs collected from www.dusit.ac.th servers. In order to improve its prediction ration, the model includes a rough sets scheme in which search similarity is measured to compute the similarity between two sequences using upper approximation.

Besides, Khasawneh and Chan [18] studied the use of a RS based learning program for predicting Web usage. In their approach, Web usage patterns are represented as rules generated by the inductive learning program, BLEM2. Inputs to BLEM2 are clusters generated by a hierarchical clustering algorithm applied to pre-processed Web log records. Their empirical results show that the prediction accuracy of rules induced by the learning program is better than a centroid based method. In addition, the use of a learning program can generate shorter cluster descriptions.

In general, the basic problems in data analysis that can be tackled using a RS approach are as follows [13]:
(i) Characterisation of a set of objects in terms of attribute values;
(ii) Finding the dependencies (total or partial) between attributes;
(iii) Reduction of superfluous attributes (data);
(iv) Finding the most significant attributes;
(v) Generation of decision rules.

According to the above basic problems, meaningless information of Web log data has the similarity problems including hidden, null and redundant situations. Hence, the main problem to tackle in this research is the decision rules generation that will be used to predict either to cache or not cache the Web documents from the meaningless to meaningful Web log data.

III. EXPERIMENTAL SETUP

This research proposes RS to reduce the rules of a log file and simultaneously enhancing the prediction performance of whether the Web object is cacheable or not [11,12]. RS is beneficial in probing the most significant attributes with crucial decision rules to facilitate intelligent Web pre-caching to safeguard limited bandwidth.

Figure 1 illustrates RS classification procedure using Rosetta System. In the first step, Web logs dataset is split into ten folds. In addition, 10-fold cross validation is implemented for validation of this experiment. The details of the 10-fold split data are presented in Table 1.

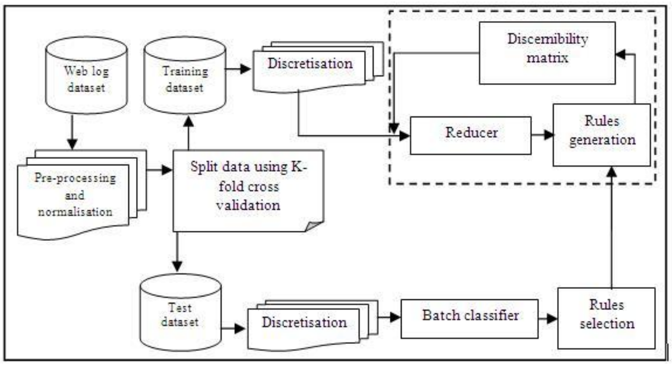

Figure 1. RS classification procedure

TABLE I. 10-FOLD CROSS VALIDATION OF BU AND EL TESTING LOG DATASET

Fold	**K-fold cross validation (10-fold)**	
	BU	**EL**
1	1-1,722	1-2,311
2	1,723-3,444	2,312-4,622
3	3,445-5,167	4,623- 6,933
4	5,168-6,890	6,934- 9,244
5	6,891-8,613	9,245- 11,555
6	8,614-10,336	11,556- 13,865
7	10,337-12,058	13,866- 16,175
8	12,059-13,781	16,176- 18,485
9	13,782-15,504	18,486- 20,795
10	15,505-17,224	20,796- 23,105

IV. Reduct and Rules Generation

The test datasets listed in the Tables 2 and 3 are labelled as K1, K2, …, K10. There are two types of reduct for discernability matrix; full reduct produces a set of minimal attribute subsets that define functional dependencies, and object related reduct produces a set of decision rules or general patterns via minimal attribute subsets that discern on a rule per object basis. The selected rules for discernibility matrix reduce more rules compared to all rules. Tables 2 and 3 depict the number of reduct and rules produced from each training dataset. BU object related reduced 4 rules more than EL object related. Moreover, K4 BU dataset produced the least number of rules for both selected and all rules. Besides, K5 BU dataset produced the most rules for each kind of reduct method. However, K1 to K10 EL dataset constructed the same number of rules for each selected and all rules reduct method, with 18 and 27 correspondingly.

These dissimilar conditions occurred because BU dataset has fewer similar attributes compared to EL dataset. These two types of Web logs were gathered from different sources. Nonetheless, EL dataset came from a specific server log data from one of E-Learning@UTM server. This means that the users access from the same e-learning website. In contrast, BU dataset is a collection of browsing history from multiple websites and diverse client workstations. The discussion on the findings will be reported in the next section.

TABLE II. NUMBER OF REDUCT AND RULES FOR BU DATASET

Discretise Method	Reduct Method	K-fold for Test Dataset	No. of Reduct	No. of Rules	Test Accuracy (%)
Naive-Bayes	GA (object related)	K1	7	1, 151	97.33
		K2	7	1, 128	97.50
		K3	7	1, 136	96.17
		K4	7	1, 097	82.41
		K5	7	1, 155	97.97
		K6	7	1, 151	96.87
		K7	7	1, 151	95.59
		K8	7	1, 136	97.91
		K9	7	1, 143	97.91
		K10	7	1, 126	96.69
	GA (full)	K1	1	2, 570	91.06
		K2	1	2, 540	91.81
		K3	1	2, 560	91.70
		K4	1	2, 510	49.45
		K5	1	2, 593	94.37
		K6	1	2, 580	91.64
		K7	1	2, 583	89.20
		K8	1	2, 531	90.83
		K9	1	2, 582	92.28
		K10	1	2, 525	90.53

TABLE III. NUMBER OF REDUCT AND RULES FOR EL DATASET

Discretise Method	Reduct Method	K-fold for Test Dataset	No of Reduct	No. of Rules	Test Accuracy (%)
Naive-Bayes	GA (object related)	K1	3	18	100.00
		K2	3	18	99.96
		K3	3	18	100.00
		K4	3	18	99.96
		K5	3	18	99.91
		K6	3	18	100.00
		K7	3	18	99.96
		K8	3	18	99.96
		K9	3	18	100.00
		K10	3	18	100.00
	GA (full)	K1	1	27	100.00
		K2	1	27	99.96
		K3	1	27	100.00
		K4	1	27	99.96
		K5	1	27	99.91
		K6	1	27	100.00
		K7	1	27	99.96
		K8	1	27	99.96
		K9	1	27	100.00
		K10	1	27	100.00

V. Decision Rules Base on Prediction

Tables 4 and 5 depict the summarisation of classification testing done on each fold of data in terms of exact result of class, prediction accuracy, error, number of reducts and rules for the corresponding dataset used. The best prediction accuracy of BU generated decision rules is 97.97% (Fold K5) and the worst is 82.41% (Fold K4). In addition, Fold K1, K3, K6, K9 and K10 of EL has the highest prediction accuracy with 100% result for each fold. From between 1, 720 and 1, 722 objects accumulated of all BU fold, 1, 6472 have been predicted successfully. Next, about 23, 099 EL objects have also been predicted fruitfully from 2, 309 to 2, 310 objects for each fold.

Figures 2 and 3 represent the relationship between prediction accuracy and number of selected rules. The results show that for BU dataset the percentage of prediction accuracy reduces when the number of selected rules decreases. For example, Fold K4 for BU dataset has the lowest percentage (82.41%) and also the most minimum number of rules, with 1, 097 rules. Nevertheless, for EL dataset, the prediction accuracy is not related with the number of success rules. The number of rules is consistent for all folds.

TABLE IV. CLASSIFICATION RESULT, PREDICTION ACCURACY, NUMBER OF REDUCTS AND RULES FOR FOLD 1 TO 10 USING BU DATASET

Fold	0	1	Error	Prediction Accuracy (%)	No. of Reduct (selected rules)	No. of Rules (selected rules)
K1	106	1, 570	35	97.33	7	1, 151
K2	85	1, 594	23	97.50	7	1, 128
K3	147	1, 510	44	96.17	7	1, 136
K4	149	1, 271	76	82.41	7	1, 097
K5	115	1, 573	28	97.97	7	1, 155
K6	132	1, 537	50	96.87	7	1, 151
K7	149	1, 497	44	95.59	7	1, 151
K8	143	1, 544	27	97.91	7	1, 136
K9	127	1, 559	25	97.91	7	1, 143
K10	180	1, 484	46	96.69	7	1, 126
Total or Average	**1, 333**	**15,139**	**398**	**95.63**	**7**	**1, 137**

TABLE V. CLASSIFICATION RESULT, PREDICTION ACCURACY, NUMBER OF REDUCTS AND RULES FOR FOLD 1 TO 10 USING EL DATASET

Fold	0	1	Error	Prediction Accuracy (%)	No. of Reduct (selected rules)	No. of Rules (selected rules)
K1	826	1, 485	0	100.00	3	18
K2	758	1, 552	1	99.96	3	18
K3	872	1, 439	0	100.00	3	18
K4	964	1, 346	1	99.96	3	18
K5	973	1, 336	2	99.91	3	18
K6	1, 155	1, 155	0	100.00	3	18
K7	1, 080	1, 229	1	99.96	3	18
K8	1, 141	1, 168	1	99.96	3	18
K9	811	1, 499	0	100.00	3	18
K10	1, 034	1, 276	0	100.00	3	18
Total or Average	**9, 614**	**13, 485**	**6**	**99.97**	**3**	**18**

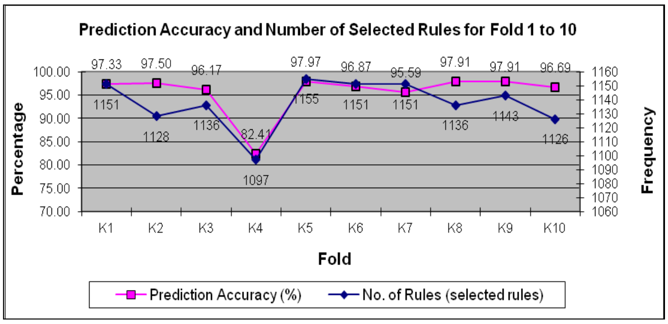

Figure 2. Prediction accuracy and number of selected rules for Fold 1 to 10 using BU dataset

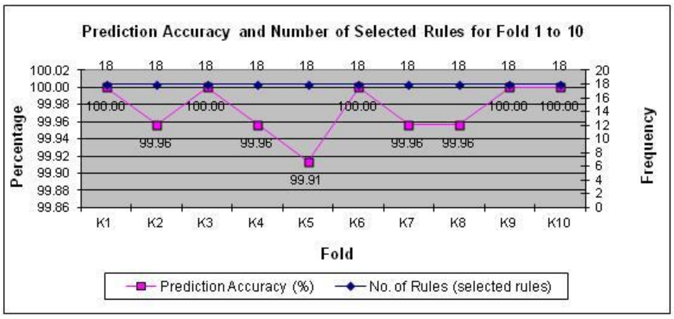

Figure 3. Prediction accuracy and number of selected rules for Fold 1 to 10 using EL dataset

Figures 4 and 5 illustrate association among class 0, 1 and error class. If a total frequency of error class is high, a frequency of either class 0 or 1 is low. For instance, the highest number of error class for BU object is 76; hence the lowest number of class 1 for BU object is 1, 271. However, the highest number of error class for EL object is 2 but neither class 0 nor class 1 has the lowest number of EL object.

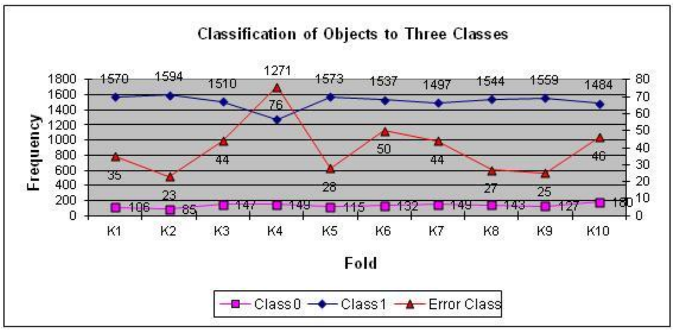

Figure 4. Classification of BU objects into three classes

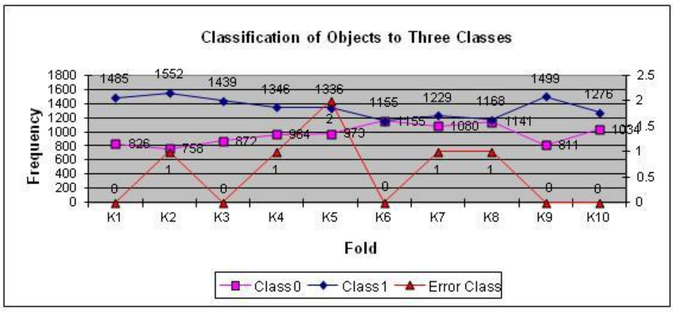

Figure 5. Classification of EL objects into three classes

VI. RULES DERIVATION AND CLASSIFICATION

Rules generation is an essential task to predict the output in RS method. A unique feature of this method is its generation of rules that plays an important role in predicting the output. Consequently, five statistical approaches are used in Rosetta's tool for the rules consisting of support, accuracy, coverage, stability and length. The definition of the rule statistics is as follows:

(i) *"The rule of LHS support is defined as the number of records in the training data that fully exhibit property described by the IF condition.*

(ii) *The rule of RHS support is defined as the number of records in the training data that fully exhibit the property described by the THEN condition.*

(iii) *The rule of RHS accuracy is defined as the number of RHS support divided by the number of LHS support.*

(iv) *The rule of LHS coverage is the fraction of the records that satisfies the IF conditions of the rule. It is obtained by dividing the support of the rule by the total number of records in the training sample.*

(v) *The rule of RHS coverage is the fraction of the training records that satisfies the THEN conditions. It is obtained by dividing the support of the rule by the number of records in the training that satisfied the THEN condition.*

(vi) *The rule of length is defined as the number of conditional elements in the IF part*." [19,20].

Tables 8 and 9 give samples of 20 from 1, 155 significant rules for fold 5 BU dataset and 18 samples of the most significant rules for Fold 1 EL dataset, which are sorted according to their support value. The most significant rule has the highest support value [19,20]. As a result, for BU and EL data decision table, the generated rule of {SIZE([0.00007, 0.66970)) => CACHE(1)} and {SIZE([0.00008, *)) => CACHE(1)} are considered as the most significant rule with the outcome of cache output equals to 1. This is supported by 10, 607 and 10, 597 support value for LHS and RHS for BU and EL data decision table, respectively. Furthermore, if the total support of BU and EL object is high, thus the coverage of both objects is also high. This situation might be proven by the highest support for both BU and EL as well as the highest coverage of BU which are 0.68428 for LHS and 0.75635 for RHS. The highest coverage for EL is 0.50962 and 0.88264 for LHS and RHS correspondingly.

Subsequently, the impact of rules length on testing accuracy were evaluated based on rules set from Tables 8 and 9. Consequently, the same rules were divided into two groups; 1≤ rules of length ≤2. It seems that the rules with LHS and RHS length ≥1 contribute better classification of BU compared to the rules with length ≤2. In other condition, the rules with RHS length of ≥1 and LHS length of ≤2 give better classification for EL data decision table. On the other hand, the RHS accuracy and stability for both BU and EL are equal to 1 for all records.

Next, Tables 6 and 7 show the overall result of classification performance of fold 1 to 10 for the original table and the new decision table of BU log file dataset. The overall classification accuracy (*C*) is based on the total of the average of each prediction accuracy (*P*) and total number of fold (*K*) as defined in equation (1):

$$C = \sum_{i=1}^{n} \frac{P_i}{K_i} \qquad (1)$$

TABLE VI. CLASSIFICATION PERFORMANCE FOR BOTH ORIGINAL DECISION TABLE AND NEW DECISION TABLE OF FOLD 1 TO 10 USING BU DATASET

Decision Table BU	**Rule Set**	**K-fold**	**Test Accuracy (%)**	**Overall Accuracy (%)**
New Decision Table	Selected Rules	K1	97.33	95.63
		K2	97.50	
		K3	96.17	
		K4	82.41	
		K5	97.97	
		K6	96.87	
		K7	95.59	
		K8	97.91	
		K9	97.91	
		K10	96.69	
Original Decision Table	All Rules	K1	91.06	87.29
		K2	91.81	
		K3	91.70	
		K4	49.45	
		K5	94.37	
		K6	91.64	
		K7	89.20	
		K8	90.83	
		K9	92.28	
		K10	90.53	

TABLE VII. CLASSIFICATION PERFORMANCE FOR BOTH ORIGINAL DECISION TABLE AND NEW DECISION TABLE OF FOLD 1 TO 10 USING EL DATASET

Decision Table EL	**Rule Set**	**K-fold**	**Test Accuracy (%)**	**Overall Accuracy (%)**
New Decision Table	Selected Rules	K1	100.00	99.97
		K2	99.96	
		K3	100.00	
		K4	99.96	
		K5	99.91	
		K6	100.00	
		K7	99.96	
		K8	99.96	
		K9	100.00	
		K10	100.00	
Original Decision Table	All Rules	K1	100.00	99.97
		K2	99.96	
		K3	100.00	
		K4	99.96	
		K5	99.91	
		K6	100.00	
		K7	99.96	
		K8	99.96	
		K9	100.00	
		K10	100.00	

Hence, Figure 6 depicts the overall accuracy for log file, with 87.29% for all rules in original decision table and 95.63% for selected rules in new decision table. This result shows a difference in overall accuracy of up to 8.34% between the original decision table and new decision table. Besides, this finding reveals that the accuracy of EL dataset is identical for both decision tables (99.97%) and gives better result compared to BU dataset. As mentioned before, one of the reasons is that BU and EL dataset are generated from different resources, client-side and server-side log files. Moreover, EL log data is specifically for E-Learning@UTM website and most of the log records are similar.

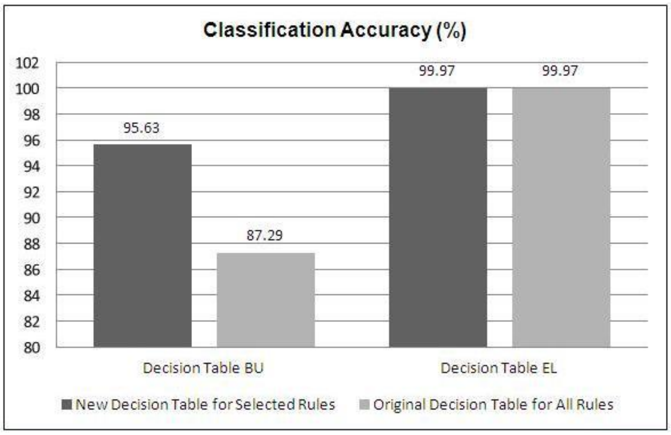

Figure 6. Overall classification accuracy for both BU and EL original decision table and new decision table of Fold 1 to 10

This result reports that RS with a new decision table is really useful for the next implementation of this collection of rules to construct queries for BU and EL objects using Social Network Analysis (SNA). In addition, the generated rules are chosen from the highest fold accuracy with the highest number of support from the new decision table.

VII. DISCUSSION

The RClass system framework [13] was used as a knowledge representation scheme for uncertainty in data to optimise the performance of proxy cache that was used to store the knowledge discovery of users' behaviours in log format. Furthermore, a substantial RS analysis based on inductive learning methods is presented to optimise Web pre-caching performance to probe significant attributes and generate the decision rules. RS granularity in WC allows decision rules to be induced [21]. These rules are important in optimising user storage by executing caching strategy in specifying the most relevant condition attributes.

VIII. CONCLUSIONS AND FUTURE WORK

This paper provides guidance to the administrator in WC regarding to selection of the best parameters to be cached and used in mobile Web pre-caching. Based on this analysis, the administrator may reorganise the parameter of log data set in proxy cache accordingly. Likewise, an empirical research has been conducted to search for the optimal classification.

In addition, RS classifier was implemented to optimise the performance of decision Web object to either cache or not cache in a proxy cache. The RS framework for log dataset was illustrated mutually with an analysis of reduced and derived rules, with entrenchment of their implicit properties for better classification outcomes. The actual prediction accuracy proves that the RS is capable to be used as a classifier in order to predict significant data from the main BU and EL datasets and will be used to produce a pull global Web pre-fetching on mobile applications. Moreover, our next phase will reveal the important of preparing and using the significant rules to identify and visualise Web objects to be cached.

ACKNOWLEDGMENT

This research is supported by The Ministry of Higher Education (MOHE) and collaboration with Research Management Centre (RMC), Universiti Teknologi Malaysia. This paper is financial supported by University Grant (GUP) Program, Tier 1 (vote: Q.J130000.2528.03H00) and FRGS Grant (vote: R.J130000.7828.4F026). The authors would like to express their deepest gratitude to the Boston University, Centre of Information and Communication Technology and Centre of Teaching and Learning, UTM especially to Mr. Azmi Kamis and Dr. Jamalludin Harun for their collaboration in making this research a successful one and K-Economy Research Alliance (RAKE), UTM for their encouragement.

REFERENCES

[1] N. Ahmad, O. Malik, , M. ul Hassan, M. S. Qureshi and A. Munir, "Reducing user latency in web prefetching using integrated techniques," *2011 International Conference on Computer Networks and Information Technology (ICCNIT)*. 11-13 July 2011, pp.175-178.

[2] L. Jian and W. Yan-Qing, "Web log data mining based on association rule," Fuzzy Systems and Knowledge Discovery (FSKD), *2011 Eighth International Conference* on (978-1-61284-180-9) Vol.3; pp. 1855-1859, 2011.

[3] C. Chunlai and L. Biwei, "A Novel Association Rules Method Based on Genetic Algorithm and Fuzzy Set Strategy for Web Mining," *Journal of Computers*, 5, 9, pp. 1448-1455, Inspec, EBSCO*host*, 2010.

[4] V. N. Padmanabhan and J. C. Mogul, "Using Predictive Prefetching to Improve World Wide Web Latency," *SIGCOMM Computer Communications Review*, 26, 3, pp. 22-36. ACM, 1996.

[5] T.S. Loon and V. Bharghavan, "Alleviating the latency and bandwidth problems in WWW browsing," In *Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems (USITS-97)*, pp. 219-230, December 1997.

[6] J. Marquez, J. Domenech, J. Gil and A. Pont, "Exploring the benefits of caching and prefetching in the mobile web," in *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, Pretoria, South Africa, 2008.

[7] S. Sulaiman, S. M. Shamsuddin and A. Abraham, "Intelligent Web Caching Using Adaptive Regression Trees, Splines, Random Forests and Tree Net," 3rd Conference on Data Mining and Optimization (DMO 2011), Malaysia, IEEE, pp. 108-114, 2011.

[8] S. Sulaiman, S. M. Shamsuddin and A. Abraham, S. Sulaiman, "Intelligent Web Caching Using Machine Learning Methods," *Neural Network World*, Volume , Issue 5, pp. 429-452, 2011.

[9] W. A. Ahmed, S. M. Shamsuddin, "Neuro-fuzzy system in partitioned client-side Web cache," *Expert Systems with Applications* 38, 2011, pp. 14715-14725.

[10] W. Ali, and S.M. Shamsuddin, "Intelligent Client-side Web Caching Scheme Based on Least recently Used Algorithm and Neuro-Fuzzy System", *The sixth International Symposium on Neural Networks(ISNN 2009), Lecture Notes in Computer Science (LNCS)*, Springer-Verlag Berlin Heidelberg , 5552, (2009), pp. 70–79.

[11] S. Sulaiman, S. M. Shamsuddin and A. Abraham, "An Implementation of Rough Set in Optimizing Mobile Web Caching Performance," Tenth International Conference on Computer Modeling and Simulation, UKSiM/EUROSiM 2008, Cambridge, UK, IEEE Computer Society Press, USA, ISBN 0-7695-3114-8, pp. 655-660, 2008.

[12] S. Sulaiman, S. M. Shamsuddin and A. Abraham, "Rough Web Caching," Rough Set Theory: A True Landmark in Data Analysis, Studies in Computational Intelligence, Springer Verlag, Germany, ISBN 978-3-540-89920-4, pp. 187-211, 2009.

[13] E. Triantaphyllou and G. Felici, "Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques," *Massive Computing Series*, Springer, Heidelberg, Germany, pp. 359-394, 2006.

[14] T. Wakaki, H. Itakura, M. Tamura, H. Motoda and T. Washio, "A Study on Rough Set-Aided Feature Selection for Automatic Web-page Classification," *Web Intelligence and Agent Systems: An International Journal* 4, pp. 431–441, IOS Press, 2006.

[15] A.H. Liang, B. Maguire and J. Johnson, "Rough Set WebCT Learning," Springer-Verlag Berlin Heidelberg, pp. 425-436, 2000.

[16] C. L. Ngo and H. S. Nguyen, "A Tolerence Rough Set Approach to Clustering Web Search Results," Springer-Verlag Berlin Heidelberg, pp. 515-517, 2003.

[17] S. Chimphlee, N. Salim, M. S. Ngadiman, W. Chimphlee and S. Srinoy, "Rough Sets Clustering and Markov model for Web Access Prediction," *Proceedings of the Postgraduate Annual Research Seminar 2006*, Malaysia, pp. 470-475, 2006.

[18] N. Khasawneh and C.C. Chan, "Web Usage Mining Using Rough Sets," 2005 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2005), pp. 580-585, 2005.

[19] I. Bose, "Deciding The Financial Health of Dot-Coms Using Rough Sets," School of Business, University of Hong Kong, 2006.

[20] N. S. Sulaiman, "Generation of Rough Set(RS) Significant Reducts and Rules for Cardiac Dataset Classification," Master thesis, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Malaysia, 2007.

[21] S. Sulaiman, S. M. Shamsuddin and A. Abraham and S. Sulaiman, "Rough Set Granularity in Mobile Web Pre-Caching," Eighth International Conference on Intelligent Systems Design and Applications - ISDA 2008, Taiwan, IEEE CS Press, USA, , pp. 587-592, 2008.

TABLE VIII. THE SAMPLE RULES FOR FOLD 5 SORTED BY THE HIGHEST RULE SUPPORT VALUES FROM BU DATA DECISION TABLE

Rule	**LHS Support**	**RHS Support**	**RHS Accuracy**	**LHS Coverage**	**RHS Coverage**	**RHS Stability**	**LHS Length**	**RHS Length**
SIZE([0.00007, 0.66970)) => CACHE(1)	10607	10607	1	0.68428	0.75635	1	1	1
NUM_OF_HITS([0.00045, *)) => CACHE(1)	5191	5191	1	0.33488	0.37015	1	1	1
RETRIEVAL_TIME([0.00297, 0.18068)) => CACHE(1)	2214	2214	1	0.14283	0.15787	1	1	1
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([0.00015, 0.00045)) => CACHE(1)	233	233	1	0.01503	0.01661	1	2	1
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([0.00015, 0.00045)) => CACHE(1)	204	204	1	0.01316	0.01455	1	2	1
RETRIEVAL_TIME([0.00253, 0.00273)) => CACHE(1)	201	201	1	0.01297	0.01433	1	1	1
RETRIEVAL_TIME([0.00232, 0.00249)) => CACHE(1)	191	191	1	0.01232	0.01362	1	1	1
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([0.00015, 0.00045)) => CACHE(1)	144	144	1	0.00929	0.01027	1	2	1
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([0.00015, 0.00045)) => CACHE(1)	139	139	1	0.00897	0.00991	1	2	1
RETRIEVAL_TIME([0.00282, 0.00296)) => CACHE(1)	112	112	1	0.00723	0.00799	1	1	1
RETRIEVAL_TIME([*, 0.00002)) AND NUM_OF_HITS([*, 0.00015)) => CACHE(0)	110	110	1	0.00710	0.07448	1	2	1
RETRIEVAL_TIME([0.00128, 0.00130)) => CACHE(1)	60	60	1	0.00387	0.00428	1	1	1
RETRIEVAL_TIME([0.00274, 0.00281)) => CACHE(1)	59	59	1	0.00381	0.00421	1	1	1
RETRIEVAL_TIME([0.00137, 0.00139)) => CACHE(1)	51	51	1	0.00329	0.00364	1	1	1
RETRIEVAL_TIME([0.00146, 0.00148)) => CACHE(1)	42	42	1	0.00271	0.00300	1	1	1
RETRIEVAL_TIME([0.00036, 0.00037)) AND NUM_OF_HITS([0.00015, 0.00045)) => CACHE(1)	42	42	1	0.00271	0.00290	1	2	1
RETRIEVAL_TIME([0.00155, 0.00157)) => CACHE(1)	41	41	1	0.00265	0.00292	1	1	1
RETRIEVAL_TIME([0.00174, 0.00176)) => CACHE(1)	40	40	1	0.00258	0.00285	1	1	1
RETRIEVAL_TIME([*, 0.00002)) AND NUM_OF_HITS([0.00015, 0.00045)) => CACHE(0)	40	40	1	0.00258	0.02708	1	2	1
RETRIEVAL_TIME([0.00107, 0.00108)) => CACHE(1)	39	39	1	0.00252	0.00278	1	1	1

TABLE IX. THE SAMPLE RULES FOR FOLD 1 SORTED BY THE HIGHEST RULE SUPPORT VALUES FROM EL DATA DECISION TABLE

Rule	**LHS Support**	**RHS Support**	**RHS Accuracy**	**LHS Coverage**	**RHS Coverage**	**RHS Stability**	**LHS Length**	**RHS Length**
SIZE([0.00008, *)) => CACHE(1)	10,597	10,597	1	0.50962	0.88264	1	1	1
SIZE([*, 0.00001)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	2,895	2,895	1	0.13922	0.32943	1	2	1
SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	2,485	2,485	1	0.11951	0.28277	1	2	1
NUM_OF_HITS([0.00132, *)) => CACHE(1)	2,092	2,092	1	0.10061	0.17425	1	1	1
SIZE([0.00002, 0.00003)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	804	804	1	0.03867	0.09149	1	2	1
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	666	666	1	0.03203	0.07579	1	2	1
SIZE([*, 0.00001)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	440	440	1	0.02116	0.05007	1	2	1
SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	381	381	1	0.01832	0.04336	1	2	1
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	361	361	1	0.01736	0.04108	1	2	1
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	279	279	1	0.01342	0.03175	1	2	1
SIZE([0.00002, 0.00003)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	172	172	1	0.00827	0.01957	1	2	1
SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0) OR CACHE(1)	132	126, 6	0.95455, 0.04546	0.00635	0.014338, 0.0005	1.0, 1.0	2	2
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	119	119	1	0.00572	0.01354	1	2	1
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	116	116	1	0.00558	0.00966	1	2	1
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	60	60	1	0.00289	0.00683	1	2	1
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	55	55	1	0.00265	0.00458	1	2	1
SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	25	25	1	0.0012	0.00208	1	2	1
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	17	17	1	0.00082	0.00142	1	2	1