Rough Set Granularity in Mobile Web Pre-Caching

Sarina Sulaiman[1], Siti Mariyam Shamsuddin[1], Ajith Abraham[2] and Shahida Sulaiman[3]

[1]*Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Johor, Malaysia*
[2]*Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway*
[3]*School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia*
sarina@utm.my, mariyam@utm.my, ajith.abraham@ieee.org, shahida@cs.usm.my

Abstract

Mobile Web pre-caching (Web prefetching and caching) is an explication of performance enhancement and storage limitation of mobile devices. In this paper, we present the granularity of Rough Sets (RS) and RS based Inductive Learning in reducing the size of rules to be stored in mobile devices. The conditional attributes such as 'Time stamp', 'Size document' *and* 'Object retrieval time' *are presented and the provided decision is granted. Decision rules are obtained using RS based Inductive Learning to grant the ultimate judgment either to cache or not to cache the conditional attributes and objects. However, in mobile clients, these rules need to be specified so as the specific sessions can be kept in their mobile storage as well as the proxy caching administrations. Consequently, these rules will notify the objects in Web application request and priority level to the clients accordingly. The results represent that the granularity of RS in mobile Web pre-caching is vital to improve the latest Web caching technology by providing virtual client and administrator feedback; hence making Web pre-caching technologies practical, efficient and powerful.*

1. Introduction

World Wide Web (WWW) has become the most ideal place for business and entertainment to enrich their presentation with interactive features. This has caused the evolution of Web, growing and rising fast and drastically. Human interaction with objects or so called interactive features has lead the Web to be more easily guided and capable to perform business task between distance places. These pages are linked and managed for certain purposes that perform as a Web application. These interactive Web pages consist of pages that are able to perform application logical task. The rising popularity of using Web applications in WWW causes tremendous demands on the Internet.

A key strategy for scaling the Internet to meet these increasing demands is to cache data near clients and thus improve access latency and reduce the network and server load [1, 2]. Caching is a technique used to store popular documents closer to the user. It uses algorithms to predict user's needs to specific documents and stores important documents. According to Curran and Duffy [3], caching can occur anywhere within a network, on the user's computer or mobile devices, at a server, or at an Internet Service Provider (ISP). Many companies employ Web proxy caches to display frequently accessed pages to their employees, as such to reduce the bandwidth at lower costs [3]. Web cache performance is directly proportional to the size of the client community. The bigger the client community, the greater the possibility of cached data being requested, hence, the better the cache's performance [3].

Prefetching is an intelligent technique used to reduce perceived congestion, and to predict the subsequently page or document to be accessed [4]. For example, if a user is on a page with many links, the prefetching algorithm will predict that the user may want to view associated links within that page. The prefetcher will then appeal the predicted pages, and stores them until the actual request is employed. This approach will display the page significantly faster compared to page request without prefetching. The only drawback is that if the user does not request the pages, the prefetching algorithm will still implement the prediction of the subsequent pages, thus causes the network to be congested [5, 6].

However, the Web caching and prefetching technologies are the most popular software based solutions [7, 4]. Caching and prefetching can work individually or combined. The blending of caching and prefetching (called as pre-caching) improves the performance by double compared to that of single caching [8]. These two techniques are useful tools to reduce congestion, delays and latency problems.

The paper is structured as follows. An overview of mobile Web prefetching and caching is presented in Section 2. In Section 3, a brief introduction of RS and inductive learning are discussed. Section 4 describes RS granularity in mobile Web pre-caching, and finally Section 5 concludes the paper.

2. Overview of mobile Web prefetching and caching (pre-caching)

Most of the existing approaches do not address the integrated scheme for mobile Web caching and prefetching. Jin *et al.* [9] presented a sequence-mining based prediction algorithm with the less amount of calculation; *mines 1-gram* association patterns from Web history records to form the prefetching candidate set for mobile client. They proposed an adaptive prefetching strategy to automatically decide the proper prefetching level in terms of mobile client's context; formulate the item's cache profit which integrates the prefetching factor. Subsequently, they devised the profit-driven caching replacement policy to glue the caching and prefetching and implement the above algorithms in Web caching system OnceEasyCache. The results were promising in various performance metrics including power, network delay, device display and size of mobile devices.

In contrast to a single algorithm, there are few works on integrated scheme. Aiming at integrated caching and prefetching, Yang and Zhang [10] employed a prediction model. Meanwhile, Teng *et al.* [11] presented a new cache replacement algorithm by considering the impact of prefetching engine located at Web server and a few cache parameters. Kobayashi and Yu [12] discussed the performance model for mobile Web caching and prefetching and provided the estimates of the total average latency, hit ratio, cache capacity and wireless bandwidth required.

Santhanakrishnan *et al.* [13] illustrated on the demand-based retrieval of the Web documents in the mobile Web. They proposed caching scheme; Universal Mobile Caching which performed the most basic and general form of caching algorithms and largely emphasize the impact of the adaptive policy. This scheme is suitable for managing object caches in structurally varying environments. Ari *et al.* [14] proposed Adaptive Caching using Multiple Experts (ACME) which the individual experts were full replacement algorithms, applied to virtual caches, and their performance was estimated based on the observed performance of the virtual caches. The term expert refers to any mechanism for offering an answer to the question. For cache replacement, the answer they seek is the identity of the object in the cache with the least likelihood of subsequent future access.

Wu *et al.* [15] introduced a rule-based modular framework for building self-adaptive applications in mobile environments. They developed techniques that combined static and dynamic analysis to uncover phase structure and data access semantics of a rule program. The semantic information is used to facilitate intelligent caching and prefetching to conserve limited bandwidth and reducing rule processing cost.

Komninos and Dunlop [16] found that calendars can really provide information to prefetch useful Internet content for mobile users. However, the anticipated problem with the current system is due to the current adaptation algorithm in adjusting the system gradually to the user needs. Thus, if a dramatic change of circumstances occurred, or if a user requested information from a very specific and known source, then it is likely that the system would fail to provide the necessary information.

In this research, rough set (RS) is employed to optimize the mobile Web pre-caching performance. RS is used to find the most significant attributes for mobile application and simultaneously to generate the decision rules.

3. Overview of Rough Sets and Inductive Learning

Rough Set Theory [17] was introduced by Zdzislaw Pawlak as a tool to solve problems with ambiguity and uncertainty [18]. Typically, data to be analyzed consists of a set of *objects* whose properties can be described by multi-valued *attributes*. The objects are described by the data that can be represented by a structure called the *information system* (*S*) [19]. An information system can be viewed as information table with its rows and columns consequent to objects and attributes.

Given a set E of examples described by an information table T, we classify objects in two different ways: by a subset C of the condition attributes and by a decision attribute D in the information table to find equivalence classes called indiscernibility classes $\Omega = \{\Omega_1,...,\Omega_n\}$ [24]. Objects within a given indiscernibility class are indistinguishable from each other on the basis of those attribute values. Each equivalence class based on the decision attribute defines a concept. We use *Des* (Ω_i) [24] to denote the description, i.e., the set of attribute values, of the equivalence class Ω_i. RS theory allows a concept to be described in terms of a pair of sets, lower approximation and upper approximation of the class. Let Y be a concept. The lower approximation $\underline{Y}$ and the upper approximation $\overline{Y}$ of Y are defined as [24]:

$$\underline{Y} = \{e \in E \mid e \in \Omega_i \text{ and } X_i \subseteq Y\} \quad (1)$$

$$\overline{Y} = \{e \in E \mid e \in \Omega_i \text{ and } X_i \cap Y = \phi\} \quad (2)$$

Lower approximation is the intersection of all those elementary sets that are contained by Y and upper approximation is the union of elementary sets that are contained by Y.

Inductive Learning is a well-known area in artificial intelligence. It is used to model the knowledge of human experts by using a carefully chosen sample of expert decisions and inferring decision rules automatically, independent of the subject of interest [20]. RS based Inductive Learning uses RS theory to find general decision rules [23, 22].

4. Rough Set granularity in mobile Web pre-caching

We used the BU Web trace dataset from Oceans Research Group at Boston University. We considered 20 sample objects only, i.e., January 1995 records. In our previous research, we used the same dataset with implementation of RS [25] and integration of Neurocomputing and Particle Swarm Optimization (PSO) algorithm [26] to optimize the Web caching performance. Three conditional attributes are taken into consideration; a number of hits (*Timestamp, TS*) in integer, a current CPU usage (*Sizedocument, SD*) in percentage and response time (*Objectretrievaltime, RT*) in seconds. Consequently, a cache, *CA* is chosen as a decision for the information table; 1 for cache and 0 for not cache. Decision rules are obtained using RS based Inductive Learning [23] for mobile Web pre-caching. Table 1 depicts the structure of the data: 20 objects, 3 attributes, and a decision.

Table 1. Sample of log files dataset information table

Object	Attributes			Decision
	TS	*SD*	*RT*	*CA*
S_1	790358517	367	0.436018	0
S_2	790358517	514	0.416329	0
S_3	790358520	297	0.572204	0
S_4	790358527	0	0	1
S_5	790358529	0	0	1
S_6	790358530	0	0	1
S_7	790358530	0	0	1
S_8	790358538	14051	0.685318	0
S_9	790362535	1935	1.021313	0
S_{10}	790362536	1804	0.284184	0
S_{11}	790362537	716	0.65038	0
S_{12}	790363268	1935	0.76284	0
S_{13}	790363270	716	1.050344	0
S_{14}	790363270	1804	0.447391	0
S_{15}	790363329	1935	0.553885	0
S_{16}	790363330	716	0.331864	0
S_{17}	790363330	1804	0.342798	0
S_{18}	790363700	0	0	1
S_{19}	790363700	0	0	1
S_{20}	790363700	1136	0.428784	0

The description and analysis of the study are given in Table 2. The domain E and two concepts Y_{cache} and $Y_{notcache}$ from the decision attribute (*CA*) are obtained as follows: $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}\}$

$Y_{cache} = \{e_4, e_5, e_6, e_{17}\}$
$Y_{notcache} = \{e_1, e_2, e_3, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{18}\}$

Initially we find the indiscernibility classes based on *TS* that are
$\{e_1,e_2\},\{e_{12},e_{13}\},\{e_{15},e_{16}\},\{e_{17}, e_{18}\}$ and
$\{e_3\},\{e_4\},\{e_5\},\{e_6\},\{e_7\},\{e_8\},\{e_9\},\{e_{10}\},\{e_{11}\},\{e_{14}\}$
The lower approximation is $\underline{Y} = \bigcup_{\Omega_i \subseteq Y} \Omega_i = \varnothing$
The upper approximation is
$\overline{Y} = \bigcup_{\Omega_i \cap Y \neq \varnothing} \Omega_i = \{e_1, e_2, e_{12}, e_{13}, e_{15}, e_{16}, e_{17}, e_{18}\}$
The discriminant index of a concept Y is defined using the following formula:

$$\alpha_{C_i}(Y) = 1 - |\overline{Y} - \underline{Y}| / |E| \quad (3)$$

Consequently, the discriminant index of *TS* is $\alpha_{C1}(Y) = 1 - |\overline{Y} - \underline{Y}| / |E| = 1 - (8 - 0)/20 = 0.6$ determines the effectiveness of the singleton set attributes consisting of *TS* in specifying the membership in *Y* (the cache concept).

Subsequently, the indiscernibility classes of *SD* is conducted and the results are $\{e_4,e_5,e_6,e_{17}\},\{e_{10},e_{12},e_{15}\},\{e_9,e_{13},e_{16}\},\{e_8, e_{11}, e_{14}\}$ and $\{e_1\},\{e_2\},\{e_3\},\{e_7\},\{e_{18}\}$.

The lower approximation is illustrated as $\underline{Y} = \bigcup_{\Omega_i \subseteq Y} \Omega_i = \{e_1\},\{e_2\},\{e_3\},\{e_7\},\{e_{18}\}$.

The upper approximation is given as

$$\overline{Y} = \bigcup_{\Omega_i \cap Y \neq \varnothing} \Omega_i = \{e_4, e_5, e_6, e_{17}, e_{10}, e_{12}, e_{15}, e_9, e_{13}, e_{16}, e_8, e_{11}, e_{14}\}.$$

Hence, the discriminant index of *SD* is $\alpha_{C2}(Y) = 1 - |\overline{Y} - \underline{Y}| / |E| = 1 - (13 - 5)/20 = 0.6$

The indiscernbility classes based on *RT* are $\{e_4,e_5,e_6,e_{17}\}$ and $\{e_1\},\{e_2\},\{e_3\},\{e_7\},\{e_8\},\{e_9\},\{e_{10}\},\{e_{11}\},\{e_{12}\},\{e_{13}\},\{e_{14}\},\{e_{15}\},\{e_{16}\},\{e_{18}\}$

The lower approximation is given as $\underline{Y} = \bigcup_{\Omega_i \subseteq Y} \Omega_i = \varnothing$. The upper approximation is $\overline{Y} = \bigcup_{\Omega_i \cap Y \neq \varnothing} \Omega_i = \{e_4, e_5, e_6, e_{17}\}$.

The discriminant index of *RT* is $\alpha_{C3}(Y) = 1 - |\overline{Y} - \underline{Y}| / |E| = 1 - (6 - 0)/20 = 0.7$

By comparing the discriminant indices of all attributes, we identify that the discriminant index of *RT* has the highest value, $\alpha_{Condition3}(Y) = 0.7$. This value determines better membership in Y. Hence, the first rule is obtained as $\boldsymbol{R_1}$**:{*Objectretrievaltime=0*}** $\Rightarrow$ **{*Cache=1*}**

Since *RT* is the most important condition attribute, we merge this condition attribute with other condition attributes to produce a new domain and to execute new rules (refer to Table 2). To discover the new domain, initially, the following equation is used to remove unnecessary elements.

$$(E - \overline{Y}) \cup (\underline{Y}) = \{e_1, e_2, e_3, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\} \cup \varnothing$$

The new element set are given as:

$$(E - [(E - \overline{Y}) \cup (\underline{Y})] = (E - \{e_1, e_2, e_3, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\} \cup \varnothing) = \{e_4, e_5, e_6, e_{17}\}$$

Subsequently, the horizontal selection of the collapsed information table is obtained (Table 3).

Table 2. Collapsed log files dataset information table

Object	Attributes			Decision	Total
	TS	*SD*	*RT*	*CA*	
e_1	790358517	367	0.436018	0	1
e_2	790358517	514	0.416329	0	1
e_3	790358520	297	0.572204	0	1
e_4	790358527	0	0	1	1
e_5	790358529	0	0	1	1
e_6	790358530	0	0	1	2
e_7	790358538	14051	0.685318	0	1
e_8	790362535	1935	1.021313	0	1
e_9	790362536	1804	0.284184	0	1
e_{10}	790362537	716	0.65038	0	1
e_{11}	790363268	1935	0.76284	0	1
e_{12}	790363270	716	1.050344	0	1
e_{13}	790363270	1804	0.447391	0	1
e_{14}	790363329	1935	0.553885	0	1
e_{15}	790363330	716	0.331864	0	1
e_{16}	790363330	1804	0.342798	0	1
e_{17}	790363700	0	0	1	2
e_{18}	790363700	1136	0.428784	0	1

Table 3. Horizontal selection of collapsed table

Object	Attributes			Decision	Total
	TS	*SD*	*RT*	*CA*	
e_4	790358527	0	0	1	1
e_5	790358529	0	0	1	1
e_6	790358530	0	0	1	2
e_{17}	790363700	0	0	1	2

The illustrations of this selected information table are given as $Y_{cache} = \{e_4, e_5, e_6, e_{17}\}$ and $Y_{notcache} = \varnothing$, and the domain is $E = \{e_4, e_5, e_6, e_{17}\}$. We locate the indiscernibility classes based on *SD* and *RT* as $\varnothing$. The lower approximation is $\underline{Y} = \bigcup_{\Omega_i \subseteq Y} \Omega_i = \varnothing$ and the upper approximation is $\overline{Y} = \bigcup_{\Omega_i \cap Y \neq \varnothing} \Omega_i = \{e_4, e_5, e_6, e_{17}\}$

The discriminant index of *SD* and *RT* is $\alpha_{C2, C3}(Y) = 1 - |\overline{Y} - \underline{Y}| / |E| = 1 - (6 - 0)/6 = 0.$

The indiscernibility classes based on *TS* and *RT* is $\{e_4, e_5, e_6, e_{17}\}$.

The lower approximation is $\underline{Y} = \bigcup_{\Omega_i \subseteq Y} \Omega_i = \{e_4, e_5, e_6, e_{17}\}$

and the upper approximation is

$$\overline{Y} = \bigcup_{\Omega_i \cap Y \neq \varnothing} \Omega_i = \{e_4, e_5, e_6, e_{17}\}$$

The discriminant index of *TS* and *RT* is

$$\alpha_{C1,\,C3}(Y) = 1 - |\overline{Y} - \underline{Y}| / |E| = 1 - (6 - 6)/6 = 1$$

By comparing the discriminant indices, we discover that $\alpha_{C1,\,C3}(Y) = 1$ best determines the membership in *Y*.

Thus, we attain the sample of second rule

***R*$_2$: {*Timestamp* = 790358527, *Objectretrievaltime* = 0} $\Rightarrow$ {*Cache* = 1}**

Two rules have been found. If new domain is uncovered and new rules are computed using the same method as previous, then the irrelevant elements can be removed as $(E - \overline{Y}) \cup (\underline{Y}) = \varnothing \cup \{e_4, e_5, e_6, e_{17}\}$

By referring to Table 3, we can see that the first set is empty and the second set has been handled by rule 2. Hence, the new set of elements becomes

$$(E - [(E - \overline{Y}) \cup (\underline{Y})] = \{e_4, e_5, e_6, e_{17}\}$$

Based on this assumption, we obtain supplementary collapsed information table in which *SD* and *RT* are omitted due to superfluous attributes (see Table 4).

Table 4. Horizontally collapsed reduction table

Object	Attributes	Decision	Total
	TS	*CA*	
e_4	790358527	1	1
e_5	790358529	1	1
e_6	790358530	1	2
e_{17}	790363700	1	2

The rules are fruitfully induced. A question that rises is how much we can believe in these rules. Therefore, we need to evaluate the strength of the rules. The strength of the rule is commonly measured [21, 19] as follows:

$$\frac{\text{\# of positive objects covered by the rule}}{\text{\# of objects covered by the rule (including both positive and negative)}}$$

Based on this equation, the first rule has a strength of 6/20. It shows that 30% Classes of e_4,e_5,e_6, and e_{17} (Table 2) are positive examples covered by the rule. Class e_1 is a negative example covered by the first rule. The second rule has the strength of 6/6, that is, 100%. In applying the first rule to this object, there is a 30% chance that the reason for cache the object is exclusively the cache of *RT*. However, there is a higher probability that the reason for cache is due to two attributes for extra timing of *TS* and *RT*, due to 100% strength of the second rule. Algorithm 1 illustrates the algorithm of rules induction using RS [23].

```
For each decision class do
Begin
  Initialise universe of objects
  Select decision class
  Find class relation
  Repeat
    For each attribute do
    begin
      Select attribute
      Find equivalence relation
      Find lower subset
      Find upper subset
      Calculate discriminant index
     end
    Select attribute with highest discriminant index
    Generate rules
    Reduce universe of objects
    Reduce class relation
  Until no objects with selected decision class
End
```

Algorithm 1. Rough set algorithm [23]

5. Conclusions

This paper presented substantial RS analysis based on Inductive Learning methods to optimize mobile Web pre-caching performance to probe significant attributes and generate the decision rules. RS granularity in mobile Web pre-caching allows decision rules to be induced. These rules are important in optimizing storage of mobile application by executing caching strategy in specifying the most relevant condition attributes. This approach provides guidance to the administrator in mobile Web pre-caching to select the best parameters to be cached. Based on this analysis, the administrator may reorganize the parameter of log data set in proxy caching accordingly.

Acknowledgments

This work is supported by MOSTI and RMC, Universiti Teknologi Malaysia, MALAYSIA. Authors would like to thank *Soft Computing Research Group* (SCRG) for their continuous encouragement and cooperation in making this study an accomplishment.

References

[1] Wong, K. Y. and Yeung, K. H., *Site-Based Approach in Web Caching Design,* IEEE Internet Comp., vol. 5, no. 5, Sept./Oct., 2001, pp. 28–34.

[2] Wong, K.Y., *Web Cache Replacement Policies: A Pragmatic Approach,* IEEE Network (Jan/Feb), 2006.

[3] Curran, K. and Duffy, C., *Understanding and Reducing Web Delays*, Int. J. Network Mgmt, 15, 2005, pp. 89-102.

[4] Garg, A., *Reduction of Latency in the Web Using Prefetching and Caching*, Doctor of Philosophy thesis, University of California, Los Angeles, United State, 2003.

[5] Yang, Q. and Zhang, Z., *Model Based Predictive Prefetching*. IEEE, 1529-4188/01, 2001, pp. 291-295.

[6] Jiang, Z. and Kleinrock, L., *Web Prefetching in a Mobile Environment*. IEEE Personal Communications, 1070-9916/98, 1998, pp. 25-34.

[7] Acharjee, U., *Personalized and Intelligence Web Caching and Prefetching.* Master thesis, Faculty of Graduate and Postdoctoral Studies, University of Ottawa, Canada, 2006.

[8] Kroeger, T. M., Long, D.D.E., and Mogul, J.C., *Exploring The Bounds of Web Latency Reduction from Caching and Prefetching*. in Proceedings of the USENIX Symposium on Internet Technology and Systems, 1997, pp. 13-22.

[9] Jin, B., Tian, S., Lin, C., Ren, X. and Huang, Y. 2007. *An Integrated Prefetching and Caching Scheme for Mobile Web Caching System.* Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp.522-527.

[10] Yang, W. and Zhang, H. H., *Integrating Web Prefetching and Caching Using Prediction Models.* Proceedings of the 10th international conference on World Wide Web, 2001.

[11] Teng, Wei-Guang, Chang, Cheng-Yue and Chen, Ming-Syan, *Integrating Web Caching and Web Prefetching in Client-Side Proxies*. IEEE Transaction on Parallel and Distributed Systems, Vol. 16, No. 5, May 2005.

[12] Kobayashi, H. and Yu, S-Z., *Performance Models of Web Caching and Prefetching for Wireless Internet Access.* International Conference on Performance Evaluation: Theory, Techniques and Applications (PerETTA 2000), 2000.

[13] Santhanakrishnan, G, Amer, A., and Chrysanthis, P.K., *Towards Universal Mobile Caching.* Proceedings of MobiDE'05, Baltimore, Maryland, USA, 2005, pp.73-80.

[14] Ari, I., Amer, A., Gramacy, R., Miller, E. L., Brandt, S., and Long, D. D. E., *Adaptive Caching Using Multiple Experts.* In Proc. of the Workshop on Distributed Data and Structures, 2002.

[15] Wu, S., Chang, C., Ho, S., and Chao, H., *Rule-based Intelligent Adaptation in Mobile Information Systems.* Expert Systems with Applications, 34(2): 2008, pp. 1078-1092.

[16] Komninos, A. and Dunlop, M.D., *A Calendar Based Internet Content Pre-caching Agent for Small Computing Devices.* Personal and Ubiquitous Computing. ISSN 1617-4909, 2007.

[17] Pawlak, Z., *Rough Sets*. pp. 3–8. Kluwer Academic Publishers, 1997.

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

[19] Johnson, J. and Liu, M., *Rough Sets for Informative Question Answering.* In Proceedings of the International Conference on Computing and Information (ICCI'98), Winnipeg, Canada, June 17-20 1996, pp. 53–60.

[20] Johnson, J. A. and Johnson, G. M., *Student Characteristics and Computer Programming Competency: A Correlational Analysis*. Journal of Studies in Technical Careers, 14: 23–92, 1992.

[21] Pawlak, Z., Grzymala-Busse, J., Slowinski, R., and Ziarko, W., *Rough sets*, Communications of the ACM, 38(11):89–95, 1995.

[22] Shan, N., Ziarko, W., Hamilton, H. J., and Cercone, N., *Using rough sets as tools for knowledge discovery.* In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'95), 1995, pp. 263–268. AAAI Press.

[23] Tsaptsinos, D. and Bell, M.G., *Medical Knowledge Mining Using Rough Set Theory,* http://citeseer.ist.psu.edu/86438.html.

[24] Liang, A.H., Maguire, B. and Johnson, J., *Rough Set Based WebCT Learning,* WAIM 2000, LNCS, 2000, pp.425-436.

[25] Sulaiman, S., Shamsuddin, S.M. and Abraham, A., *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, 2008, pp. 655-660.

[26] Sulaiman, S., Shamsuddin, S.M., Forkan, F. and Abraham, A., *Intelligent Web Caching Using Neurocomputing and Particle Swarm Optimization Algorithm*, Second Asia International Conference on Modeling and Simulation, AMS 2008, IEEE Computer Society Press, USA, 2008, pp. 642-647.