Usability Study on Mobile Web Pre-Fetching

Sarina Sulaiman, Nadia Lokman and Nur Syahela Hussien
UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research,
Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia.
sarina@utm.my

Abstract**—Web pre-fetching is a technique that can increase the speed of web loading process. It refers to the operation of fetching information from remote web servers even before it is requested. This research approves a web pre-fetching method that is capable to decrease user-perceived latency in web browsing for mobile environment, in which it is essential as mobile devices have several limitations compared to normal desktop PC. Advanced Keystroke Level Model (KLM) technique was used where it predicted the interaction time of user with Mobile Web Pre-Fetching (MWeP) prototype and it is validated through the statistical method of single factor controlled experiments where user testing was done. Facebook was chosen as the case study where the usability evaluation based on advanced KLM technique supported that user access time with MWeP prototype decreased considerably while the usability evaluation through user testing implied that the mobile web pre-fetching prototype was satisfactory to user in terms of learnability, efficiency of use, memorability, error frequency and satisfaction. This research indicated that the utilisation of web pre-fetching is recommended in mobile environment as it effectively decreases user interaction time during web browsing that enhances user's overall browsing experience. Based on advanced KLM technique, the total access time for mobile application with MWeP method decreased than those without MWeP.**

Index Terms**—Advanced Keystroke Level Model; Mobile Web Pre-Fetching; Mobile Environment; Usability Study.**

I. INTRODUCTION

The use of World Wide Web through mobile applications had increased in popularity in recent years [1]. However, with limited resources like speed and memory that mobile devices have, user tends to perceive latency in their browsing experience [2]. In mobile web applications, the latency issues include the factor of it to handle large web objects efficiently that usually contains many resources such as CSS, JavaScript, images and videos developed mainly for PC desktop web experience. Furthermore, the high network latencies of mobile phones can also make mobile web browsing a frustrating experience [3].

Web pre-fetching is one of the ways to handle these issues where it is an effective scheme to lessen the service bottleneck, minimise user access latency and reduce network traffic [4]. Web pre-fetching could increase the speed of web loading process by fetching information from remote web servers even before the user requests for it [5].

The study on usability of mobile web pre-fetching is done to evaluate whether it is relevant to be applied in mobile environment, where user interaction time with mobile web applications that do and do not apply web pre-fetching method is calculated by using the Advanced Keystroke Level Model (KLM) technique and its validation through user testing.

II. WEB PRE-FETCHING IN MOBILE ENVIRONMENT

With increased growth of mobile users that often access the World Wide Web nowadays, it makes it necessary for users to be able to experience fast browsing in mobile environment. Thus, the necessity of mobile web pre-fetching is justified; in which several existing literatures have discussed the various techniques of web pre-fetching that is considered suitable to be applied in the mobile environment.

Jiang *et al.* [6] proposed web pre-fetching in mobile environment where it uses an adaptive pre-fetch scheme. In this approach, pre-fetch decision is able to adapt to different network conditions that requires two modules of prediction and pre-fetch module. These modules compute the access probabilities and pre-fetch thresholds respectively. The access probabilities indicate how likely files will be requested by the user and the pre-fetch thresholds determine whether the performance may be improved by pre-fetching certain files. As user changes network in a mobile environment, it is the pre-fetch threshold that is computed. The computation is based on system conditions as well as costs of bandwidth and time that adjusts the number of pre-fetch files. This approach is based on user's access history, so it is unable to predict a new web object.

Sulaiman et al. [7] on the other hand, proposed a technique known as an Intelligent Mobile Web Pre-fetching (IMWeP) that creates pre-fetching in a mobile environment using the Extensible Markup Language (XML). The system calculates the number of hits for each mobile web application features during user's browsing activity. It will automatically classify user's most favorite features from a website, before recording each activity and restoring the activity during the second login. If pre-fetched files exist, IMWeP will directly guide user to the appropriate web application features. This web pre-fetching method in IMWeP was adapted into this research as a prototype called Mobile Web Pre-fetching (MWeP).

III. ADVANCED KEYSTROKE LEVEL MODEL (KLM)

The Keystroke Level Model (KLM) was proposed to predict the task execution time from a specified design and specific task scenario [8]. It is an approach of human-computer interaction as a method for predicting user performance where the execution time is estimated by listing the sequence operators before summing the times of the individual operators.

The original KLM had six classes of operators which are K that is for pressing a key, P for pointing to a location on screen with the mouse, H for moving hands to home position on the keyboard, D for drawing straight line with the mouse, M for mentally preparing to perform an action, and R for system response where the user waits for the system. For each operator, there is an estimate of execution time.

Although the basic KLM is considered competent, but as the interaction between user and mobile devices is slightly different than those of desktop, Advanced KLM technique is utilised in the usability evaluation for mobile web pre-fetching as it has few added new operators that defines the interactions of user with mobile devices more clearly [9].

The new operators are S_{Macro} for attention of user between the contents on the screen of the mobile device to an object, S_{Micro} for looking from the display to the keypad and hotkey regions and vice versa, X for distraction, A for executing a certain complex action with the phone, G for mobile phone gestures, F for finger movement, and I for initial act. Although the Advanced KLM adapts mostly all of the action operators from the basic KLM, there are some that may be of same operators but with altered estimate of execution time. Table 1 lists the actions and execution time estimation for the Advanced KLM.

Table 1
Actions and Execution Time Estimation for Advanced KLM Technique [9]

Operator	Remarks	Time (s)
A	Certain interactions require additional actions	1.23
I	Find and start interaction with device (initial act)	
	Externally	5.32
	Internally	3.89
	Optimal setting	1.18
	No assumptions	4.61
F	Finger movement on a small screen	0.23
G	Gesture interaction with finger, hand, etc.	0.80
K	Press key	
	Good typist	0.33
	Average typist	0.39
B	Mouse button	
	Press or release	0.10
	Click	0.20
P	Moving mobile device	1.00
H	Home hands to and from keyboard	0.95
M	Mental preparation for a subtask	1.35
S_{Macro}	Attention shift; world ↔ device	0.36
S_{Micro}	Attention shift; display ↔ keypad and hotkey regions	0.14

The design of applications for mobile devices needs a different quality assessment than those of desktop applications. Holleis *et al.* [9] stated many aspects that have to be taken into account; one important criterion is the average time user needs to complete a task. For usability evaluation of mobile web pre-fetching, the interaction times could be predicted by using Advanced KLM in which the mobile web pre-fetching can be evaluated with respect to user performance time.

IV. Usability Testing

Usability testing is a technique to evaluate a system by testing it on user, which gives direct input on how real user uses the system [10]. In this research, usability evaluation of mobile web pre-fetching was done by measuring the user interaction time with the pre-fetched mobile web application. Usability is a combination of factors including intuitive design, ease of learning, efficiency of use, memorability, error frequency and severity, and user's satisfaction. For this research, all the aspects will be taken into account to compare usability between normal mobile web browsing and with web pre-fetching method.

The number of test users for the usability study was only five people as recommended by Nielson. According to the proposed method, five people as test users is sufficient as it assents to find almost as many usability problems as would be by using many more test participants [11]. Based on Figure 1, it can be seen that testing more users did not result in appreciably more insights as the line across the chart which represents the number of usability findings is generally similar either with five or more participants. This research focused on the study of the usability of web pre-fetching in mobile environment to oversee mobile web pre-fetching efficiency through formative evaluation of Advanced Keystroke Level Model (KLM).

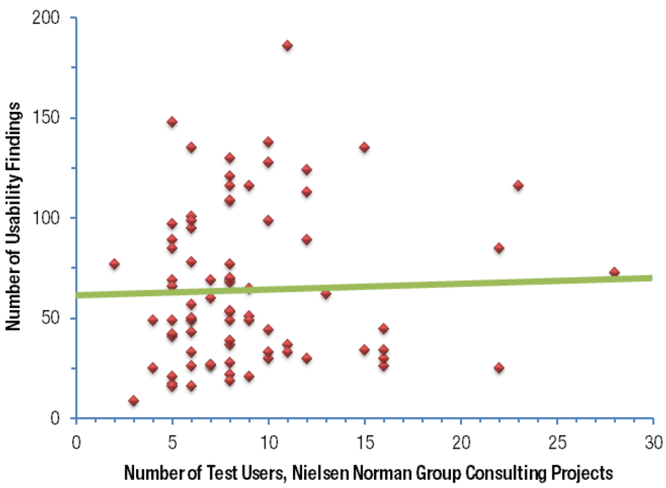

Figure 1: Usability findings proportional to number of test user by Nielson [11]

V. Result Analysis and Discussion

A. Usability Evaluation of Mobile Web Pre-Fetching Through Advanced Keystroke Level Model (KLM) Technique

In evaluating the usability of mobile web pre-fetching, the Advanced KLM technique was applied through three scenarios which consist of the normal web application browsing without using pre-fetching and those that use pre-fetching, either with or without first time access to the MWeP prototype. Facebook was chosen as the case study, where each user interactions and actions with Facebook features were calculated using the Advanced KLM approximation time (refer to Table 1).

For the first scenario, user logs in to Facebook without web pre-fetching technique where user needs to open an Internet browser first. Then, user needs to insert its user name and password before the home feature appears. After that, user will continue to browse through other features to their liking. Table 2 demonstrates in detail the actions and total time taken by user for this scenario.

For the second scenario, user logs in for the first time in MWeP prototype where it applies the web pre-fetching technique. User needs to complete a number of steps similar as in the first scenario, but for the second and the following logins, user is only required to fill in the user name and password in the prototype before being directed to link feature that has the highest number of hit. The actions and total time for the second scenario are as in Table 3.

Table 2
Actions and Total Time for Scenario without Mobile Web Pre-fetching (Scenario 1)

Action Sequence	Operator Sequence	Calculation of Total Time
1. Initial act	1. Start using mobile phone, I	= I+6F+8K+5K+7K+3B
2. Point to browser application	2. Finger move to browser application, F	= 1.18+(6*0.23)+(8*0.39)+(5*0.39)+(7*0.39)+(3*0.20)
3. Click browser application	3. Click browser application, B	
4. Point to URL text box	4. Finger move to URL link box, F	
5. Write the URL link	5. Write the URL link (8 letter word), K	
6. Point to username text box	6. Finger move to username text box, F	= 1.18+1.38+3.12+1.95+2.73+0.60
7. Write username	7. Write username (assume 5 letter word), K	= 10.96 seconds
8. Point to password text box	8. Finger move to password text box, F	
9. Write password	9. Write password (assume 7 letter word), K	
10. Point to log in button	10. Finger move to log in button, F	
11. Click log in button	11. Click log in button, B	
12. Point to message feature	12. Finger move to message feature, F	
13. Click message feature	13. Click message feature, B	

Table 3
Actions and Total Time for Scenario with Mobile Web Pre-fetching for First Time Log In (Scenario 2)

Action Sequence	Operator Sequence	Calculation of Total Time
1. Initial act	1. Start using mobile phone, I	= I+5F+5K+7K+3B
2. Point to MWeP application	2. Finger move to MWeP application, F	= 1.18+(5*0.23)+(5*0.39)+(7*0.39)+(3*0.20)
3. Click MWeP application	3. Click MWeP application button, B	
4. Point to username text box	4. Finger move to username text box, F	
5. Write username	5. Write username (assume 5 letter word), K	= 1.18+1.15+1.95+2.73+0.60
6. Point to password text box	6. Finger move to password text box, F	
7. Write password	7. Write password (assume 7 letter word), K	=7.61 seconds
8. Point to log in button	8. Finger move to log in button, F	
9. Click log in button	9. Click log in button, B	
10. Point to message feature	10. Finger move to message feature, F	
11. Click message feature	11. Click message feature, B	

The third scenario allows user to directly navigate the tree view menu that exists in the MWeP prototype. The tree view consists of user's link features hit ratio, whereby the most visited feature will be at the top of the node. Table 4 remarks the details of actions and total time taken by user for the third scenario, where user will firstly open the tree view element and check data information before clicking on a feature that they want to visit.

Table 4
Actions and Total Time for Scenario with Mobile Web Pre-fetching (Scenario 3)

Action Sequence	Operator Sequence	Calculation of Total Time
1. Initial act	1. Start using mobile phone, I	= I+3F+3B
2. Point to MWeP application	2. Finger move to MWeP application, F	= 1.18+(3*0.23)+(3*0.20)
3. Click MWeP application	3. Click MWeP application button, B	
4. Point to tree view feature	4. Finger move to tree view feature, F	= 1.18+0.69+0.60
5. Click tree view feature	5. Click tree view feature, B	= 2.47 seconds
6. Point to message feature	6. Finger move to message feature, F	
7. Click message feature	7. Click message feature, B	

Based on the usability evaluation that had been done, it could be seen that accessing mobile web application without web pre-fetching took more time compared to those that used mobile web pre-fetching technique. Mobile web application that did not use pre-fetching took 10.96 seconds to access, while mobile web application that uses web pre-fetching took 7.61 seconds for first time log in, and 2.47 seconds with the tree view feature. Figure 2 illustrates the graph of access time comparison between the three different scenarios.

The results of this evaluation implies that web pre-fetching in mobile environment is essential as it shows a decrease in user's access time for up to 77.46% in an optimum setting (refer to Equation 1). With many links and features that exist in a website, MWeP pre-fetching which shows five most top features that user visits made it convenient for user. This is especially true as for example, Facebook itself has over than ten pages.

$$\%\ \text{decrease} = \frac{(\text{original number} - \text{new number})}{\text{original number}} * 100 \quad (1)$$

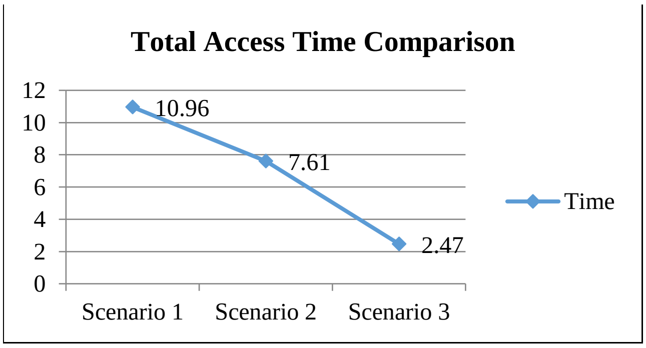

Figure 2: Total access time comparisons between mobile web applications with and without MWeP

B. Usability Evaluation of Mobile Web Pre-Fetching Through User Testing

The usability evaluation for this research focuses on the learnability, efficiency of use, memorability, error frequency and satisfaction aspect of the mobile web pre-fetching application prototype (MWeP). Learnability is how easy it is for users to accomplish tasks the first time they use a system, efficiency is how quickly users could perform tasks in the system, memorability is how easily can users use the system the second time around, while error frequency is how many errors do users make while using the system, and satisfaction is how pleasant it is for users to use the system. Five users were selected to first employ the conventional or normal mobile web browsing of Facebook, and then test the prototype and were required to complete a questionnaire to rate the applications from the scale of 1 to 5 which represents strongly disagree, disagree, moderately agree, agree or strongly agree. The obtained data will subsequently be calculated to find its mean, median and standard deviation.

For the learnability aspect, users were asked whether is it easy to navigate through and accomplish tasks the first time they browse Facebook either using normal browser or with web pre-fetching method (MWeP). Figure 3 demonstrates that the mean for normal browsing is 3.8 with median of 4 and standard deviation of 0.98, while browsing through MWeP prototype is 4.4, 5 and 0.8 for its mean, median and standard deviation. This result shows that users have no problems in using MWeP prototype for the first time as the mean and median score of 4.4 and 5 demonstrates the statement of strongly agree compared to normal browsing which has the mean of 3.8 which also indicates smooth navigating for the first time.

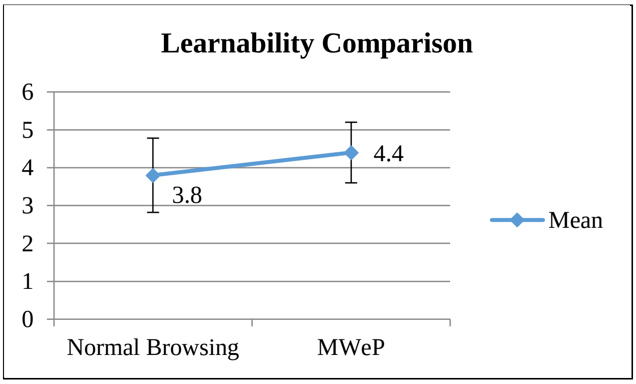

Figure 3: Learnability evaluation comparisons for normal browsing and with MWeP prototype

For the evaluation of efficiency of use, users said that normal browsing is moderate in efficiency with mean and median of 3 and standard deviation of 0.89. MWeP prototype on the other hand, is considered efficient by users where most of them strongly agree with its efficiency of use where it scores the mean of 4.6, median of 5 and standard deviation of 0.49. These results show that users experienced an improvement in overall browsing through MWeP prototype as the evaluation scale of 1 to 5 increases by 2. Figure 4 illustrates the results for this test.

For the memorability evaluation of normal browsing and with web pre-fetching technique, users were inquired whether they can remember how to use the web applications effectively in future visits. Figure 5 shows that normal browsing receives the score of 4 for its mean and median, and 0.63 for its standard deviation. MWeP prototype similarly gains the same ratings of 4.2 for its mean, 4 for its median and 0.75 for its standard deviation. This result demonstrates that users do not have a clear distinction between normal browsing and those with web pre-fetching technique as the two results are just slightly different. Thus, it can be said that the two methods of browsing are quite straightforward and users could easily remember and navigate through it the second time around.

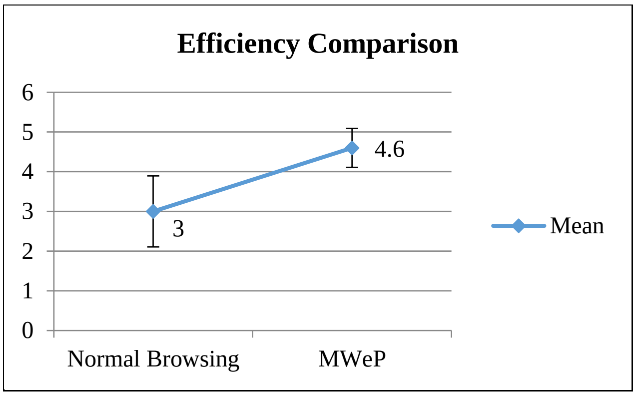

Figure 4: Efficiency evaluation comparisons for normal browsing and with MWeP prototype

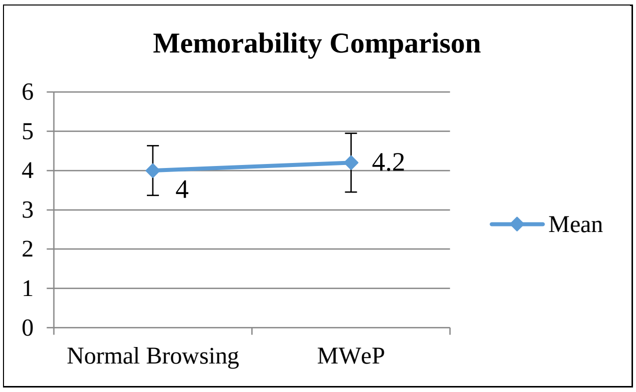

Figure 5: Memorability evaluation comparisons for normal browsing and with MWeP prototype

In the evaluation of error frequency, users were asked whether they often make mistakes or error while using either the web pre-fetching prototype or normal browsing. Figure 6 reveals that normal browsing has the mean of 3.4, median of 3 and standard deviation of 1.02. This indicates that users moderately encounter errors throughout normal browsing while occasionally through MWeP browsing. This is because the evaluation of MWeP prototype has the mean of 2.4, median of 2 and standard deviation of 0.49 which is just slightly lesser in the scale of evaluation compared to normal mobile web browsing.

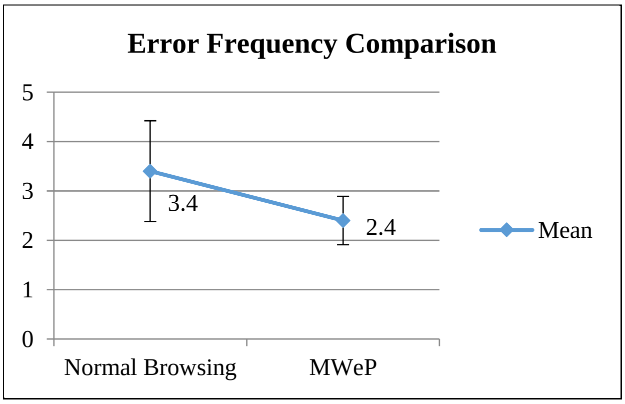

Figure 6: Error frequency evaluation comparisons for normal browsing and with MWeP prototype

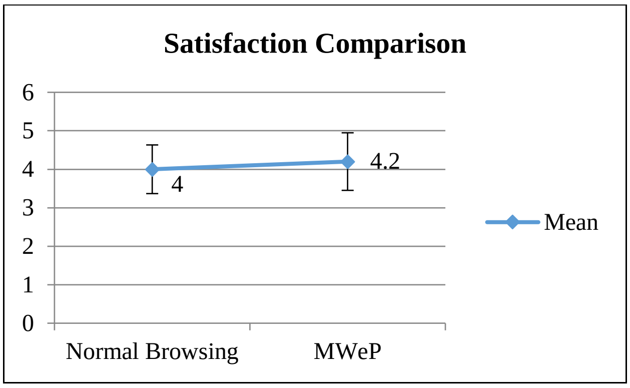

Figure 7: Users' satisfaction evaluation comparisons for normal browsing and with MWeP prototype

As for the satisfaction aspect, users seemed content with the use of both mobile web applications as shown in Fig.7. The figure shows that MWeP prototype browsing is just slightly better than the normal browsing in terms of users' overall satisfaction with a mean of 4.2, median of 4 and standard deviation of 0.75. Normal mobile web browsing on the other hand attains mean and median of 4 and standard deviation of 0.63.

Based on the usability evaluation of both normal mobile web browsing and web pre-fetching, most of the results are in favor of supporting the use of MWeP, where the tendency of users based on the standard deviations is close to 0, in which the data point does not spread out over a wide range of values. This situation indicates that the results are very close to the mean or the expected values of the sets that are within 4 to 5 (agree and strongly agree).

VI. Conclusion and Future Work

In conclusion, web pre-fetching is deemed suitable to be implemented in mobile environment as it decreases the access time during web browsing where in the usability evaluation through Advanced KLM technique, it has been shown that the total access time for mobile application with MWeP method is lesser than those without MWeP. Furthermore, usability evaluation through user testing implies that the MWeP prototype is mostly satisfying to users, with the range of scale that is slightly higher compared to normal mobile web browsing. For future works, the web pre-fetching method for mobile environment should be implemented in a generic manner where it does not apply to only specific websites and mobile applications. User should be able to visit any websites they want without any constraint and the system should be able to also pre-fetch the most favorite websites that user visits, alongside its top features. Furthermore, the web pre-fetching method in mobile environment should combine web caching on a proxy in the future work.

Acknowledgment

This research is supported by The Ministry of Higher Education (MOHE), Ministry of Science, Technology and Innovation (MOSTI) and collaboration with Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM). This work is supported by Ministry of Higher Education Malaysia (MOHE), Ministry of Science, Technology and Innovation Malaysia (MOSTI), and Universiti Teknologi Malaysia (UTM). This paper is financially supported by E-Science Fund, R.J130000.7928.4S117, PRGS Grant, R.J130000.7828.4L680, GUP Tier 1 UTM, Q.J130000.2528.13H48, FRGS Grant, R.J130000.7828.4F634 and IDG Grant, R.J130000.7728.4J170. The authors would like to express their deepest gratitude to the Smart Digital Community Research Alliance, UTM and Soft Computing Research Group (SCRG) for their encouragement.

References

[1] S. Teltscher, E. Magpantay, , I. Vallejo, L. Kreuzenbeck, D. Korka, V. Gray, and D. Olaya, *Measuring the Information Society*. Geneva: International Telecoms Union, 2013.

[2] S. Sulaiman, S.M. Shamsuddin, and A. Abraham, "Rough neuro-PSO web caching and XML pre-fetching for accessing Facebook from mobile environment," in *World Congress on Nature and Biologically Inspired Computing*, 2009, pp. 884-889.

[3] D. Lymberopoulos, O. Riva, K. Strauss, A. Mittal, and A. Ntoulas, "PocketWeb: Instant web browsing for mobile devices," in *ASPLOS XVII Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2012, pp. 1-12.

[4] J. A.K. Sharma, and A. Goel, "A Framework for prefetching relevant web pages using Predictive Prefetching Engine (PPE)," *IJCSI International Journal of Computer Science*, vol. 7, no. 6, pp. 1-9, 2010.

[5] R. Suryawanshi, and A. Savyanavar, "Web caching and XML prefetching for accessing social sites from mobile environment," *International Journal of P2P Network Trends and Technology*, vol. 2, no. 1, pp. 38-41, 2012.

[6] Y. Jiang, and L. Kleinrock, "Web pre-fetching in a mobile environment," *IEEE Personal Communications*, vol. 5, no. 5, pp. 25-34, 1998.

[7] S. Sulaiman, S.M. Shamsuddin, A. Abraham, and S. Sulaiman, "Intelligent mobile web pre-fetching (IMWeP) using XML technology," in *6th IEEE International Conference on Next Generation Web Services Practices*, 2010, pp. 475-480.

[8] D. Kieras, "Using the keystroke-level model to estimate execution times," in *The Computer Science and Engineering Handbook*, A.B. Tucker, Ed. CRC Press, Boca Raton, 1993.

[9] P. Holleis, F. Otto, H. Hubmann and A. Schmidt, "Keystroke-level model for advanced mobile phone interaction," in *CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2007, pp. 1505-1514.

[10] J. Nielsen, "Usability 101: Introduction to Usability", 2012. [Online]. Available at https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[11] J. Nielsen, "How Many Test Users in a Usability Study", 2012. [Online]. Available at https://www.nngroup.com/articles/how-many-test-users/