Student Classification in Adaptive Hypermedia Learning System Using Neural Network

Bariah Yusob[a], Nor Bahiah Hj Ahmad[a], Shahliza Abd Halim[a], Norazah Yusof[a] and Siti Mariyam Hj Shamsuddin[b]

[a]Department of Software Engineering
Faculty of Computer Science and Information System
University of Technology Malaysia, 81310 Skudai
Tel: 07-5532356, Email: bahiah@fsksm.utm.my

[b]Department of Graphic and Multimedia
Faculty of Computer Science and Information System
University of Technology Malaysia, 81310 Skudai
Tel: 07-5532321, Email: mariyam@fsksm.utm.my

ABSTRACT

Conventional hypermedia learning system can pose disorientation and lost in hyperspace problem that will cause learning objectives hard to achieve. Adaptive hypermedia learning system is the solution to overcome this problem by personalizing the learning module presented to the student based on the student knowledge acquisition. This research aims to use neural network to classify the student whether he is advanced, intermediate and beginner student. The classification process is important in adaptive hypermedia learning system in order to provide suitable learning module to each individual student by taking consideration of the students' knowledge level, his learning style and his performance as he learn through the system. Data about the student will be collected using implicit and explicit extraction technique. Implicit extraction technique gathers and analyses the student's behavior captured in the server log while they navigate through the system. Explicit extraction technique on the other hand collects student's basic information from user registration data. Three classifiers were identified in determining the student's category. The first classifier determines the student initial status based on data collected from explicit data extraction technique. The second classifier identifies student's status from implicit data extraction technique by monitoring his behavior while using the system. The third classifier, meanwhile will be executed if the student has finished studying and finished doing the exercises provided in the system. Further, the data collected using both techniques will be integrated to form a user profile that will be used for classification using simple back propagation neural network.

Keywords
Classification, web log analysis, neural network, adaptive hypermedia learning system, personalization.

1.0 INTRODUCTION

Adaptive hypermedia learning system (AHLS) is an approach to overcome the problems with traditional static hypermedia learning system (Brusilovsky, 1994). AHLS manage to present interactive and dynamic interface to provide suitable learning module to each student. Learning module presented is based on students' knowledge and skill level together with their preferences that can be seen through analysis of their behavior as they navigate along the system.

Learning module presentation that meets those features must be created through one very popular concept: personalization (Wang, 2000). This concept has been used as a trend in electronic application in World Wide Web including e-commerce, e-medicine and others. Through personalization, users are no longer treated equally. AHLS will recognize users and provide services according to their personal needs. For example, a beginner student will not be presented with learning module that is too complex and have difficult paths. If the student receive unsuitable module that does not match his level, he will not be able to follow the learning process and finally the objective of the module is not achieved.

To apply personalization, student's information must be extracted and analyzed to get useful information about the student and to identify the type of the student, whether he is advanced, intermediate or beginner student. The aim of this project is to integrate two types of data extraction techniques in order to develop user profile. Data about the user can be extracted explicit and implicitly. Explicit technique includes data given by the students when they use the system such as, by fill in their personal information during registration and the total score they gain as they perform exercises after finish certain topic. Information about the user can be extracted implicitly by monitoring the students' behavior while they interact with the system. At this stage, the students are not aware that their movements are recorded in the web log file. Data collected in the server log are navigation paths, click streams, together with the date and time the page is requested.

This paper presents the preliminary work in identifying the students' category by manipulating the data extracted explicitly and implicitly. We used data stored in the system for the neural network training to perform the classification task. This paper will first present a model of student classification using neural network. In the second section we will describe the process of analyzing students' activity in web log data, including navigation, number of using help and number of backtracking. There are three types of classifier for determining the student's status which will be explained in detail in the subsequent section.

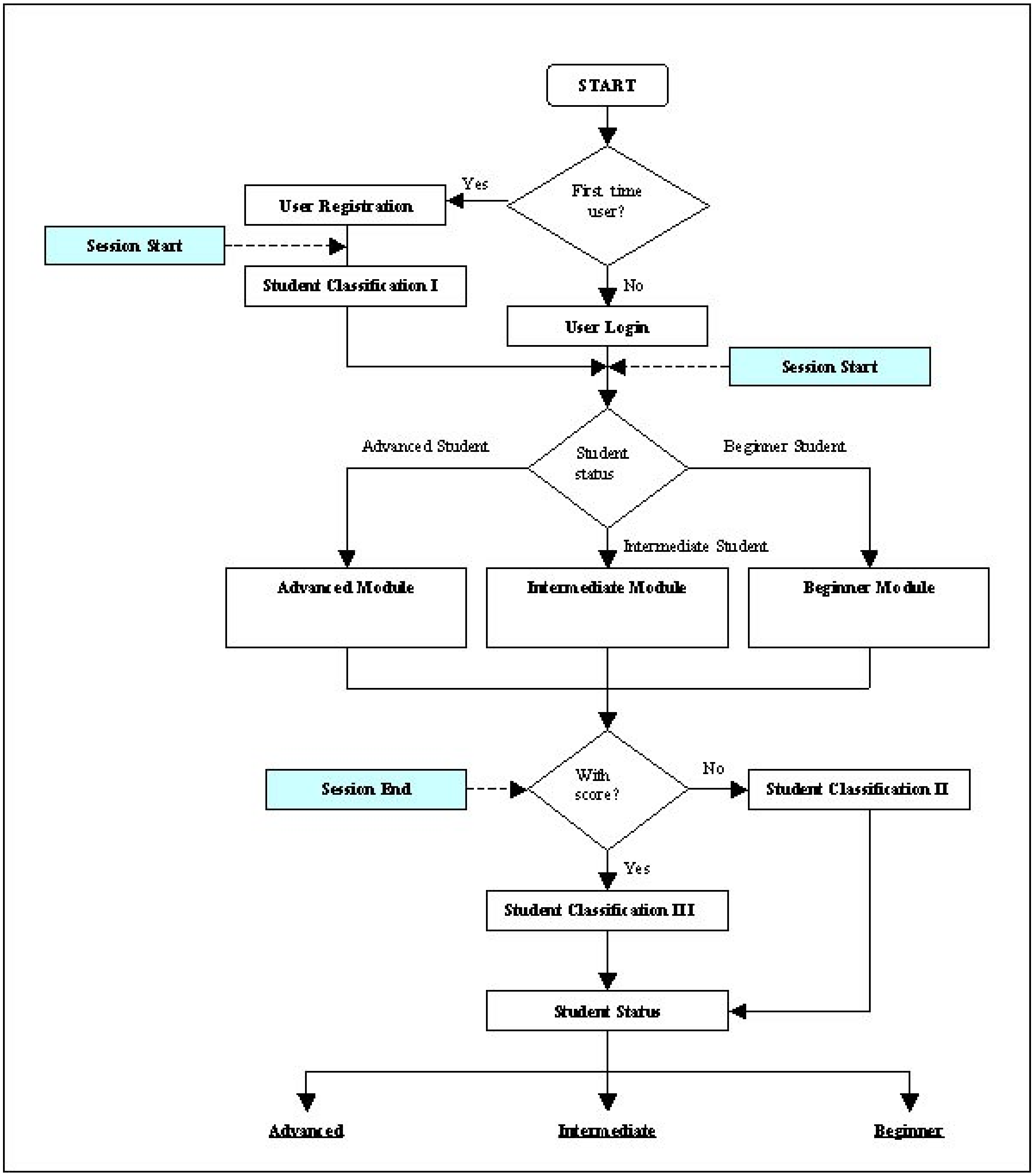


Figure 1: Workflow of Student Classification

2.0 WORKFLOW OF STUDENT CLASSIFICATION USING NEURAL NETWORK

Figure 1 shows that the system must first identify whether the student has registered into the system before allowing them to start learning the module. For the classification purpose, there are three situations need to be considered when the student access the system. The situations include whwther the student is first time user, student without score value and student with score value. In the following, we will give an overview of each situation:

i) Student who has not register into the system. It means that the student is a new user. The student must enter his information including his name, metric card number, identification card number (ic), cumulative point average (cpa) and programming knowledge (prog). Once the student submit his registration data, the system will record the data into database and remember his metric number as a new user. Then, the system will execute the first classifier to identify his initial status.

ii) The student has registered in the system and will be asked to login into the system by entering his metric number so that the system could recognize him and will retrieve his previous status. Based on his previous status while using the system before, the suitable learning module will be presented. In this situation, there is a possibility that the student simply finish learning the module without performing any exercises. Once he log out of the system, the second classifier will identify his new status.

iii) The student who has registered into the system, has finished learning the module and has completed doing exercises. The score for the exercises will be stored in the system database. The third classifier will identify his new status using explicit and implicit data extraction.

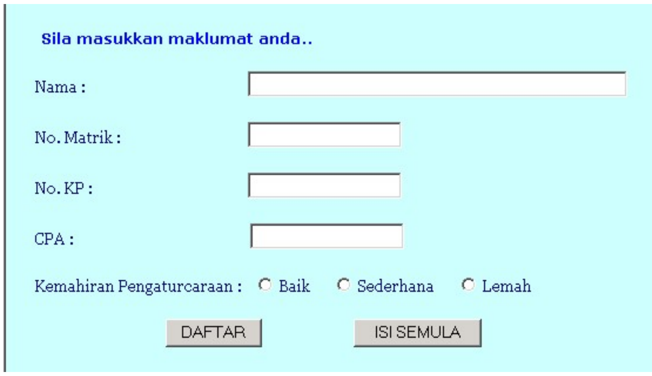
For situation two and situation three, the system must first identify students' status from the previous learning session. Once their status is identified, the learning module will be presented according to their status. After they finished their session, the last classifier will be applied to categorize them into their current status to be used in the next session when they come back for learning new module.

3.0 EXPLICIT DATA EXTRACTION: CLASSIFIER I

First time user, the user who has never use the system, must register into the system by filling his personal information into the student registration form provided in the system. Explicit data is the data provided interactively and willingly by the student.

3.1 Student Registration

Data collected from student registration process are student's name, metric card number, identity card number, cumulative point average and the student's programming knowledge level. Figure 2 shows the student registration form. Registration data will be stored in the database. Once the student submit his personal information the system will assign a new session that will recognize the student by his metric number, noMatrik. Data collected from the registration will be used in student classification phase to identify the initial status of the student.



Sila masukkan maklumat anda..

Nama :

No. Matrik :

No. KP :

CPA :

Kemahiran Pengaturcaraan : ○ Baik ○ Sederhana ○ Lemah

Figure 2: Student Registration Form

3.2 Training Data Sample

Before data can be fed into neural network for student classification, one set of training data is needed. Training data is used to train the network to perform classification into desired groups. To obtain the training sample, one program is developed to calculate the desired output values. The input data for the program are cumulative point average (cpa) and programming knowledge (prog).

Table 1: Input Data for Classifier I

Input, x	Value	Weight, w
cpa, x_1	0.00 – 4.00	$w_1 = 0.7$
prog, x_2	1,2,3	$w_2 = 0.3$

Where,

$$\text{Output, } y = x_1 * w_1 + x_2 * w_2$$

Student's cumulative point average, cpa, will hold the value between 0.00 – 4.00, while programming knowledge level, prog, will hold the value 1 for beginner, 2 for intermediate and 3 for advanced. The weight is given based on the priority between these two data. Based on our observation, cpa carries more priority than prog value.

Classification I:

```
If $y > 2.00$ then
        Status = beginner
Else if $2.00 \leq y \leq 3.00$ then
        Status = Intermediate
Else if $y > 3.00$ then
        Status = advanced
```

Data Representation:

Beginner	: 00
Intermediate	: 01
Advanced	: 11

4.0 IMPLICIT DATA EXTRACTION: CLASSIFIER II

Implicit data extraction is a process of analyzing web log data, which contain students' activity through their interaction with the system.

4.1 Web Log Analysis

Web log analysis is a process of extracting useful information about user's behavior recorded in web log server file. In this project, we perform implicit technique by collecting web log data and analyze it to get students' path navigation through the system. Figure 3 shows lines from a file which contains the following fields: date, time, client IP address and URI stem (requested node). The original data show sequences of requests by IP address and request time. In order to analyze the log data, we first need to define an individual session.

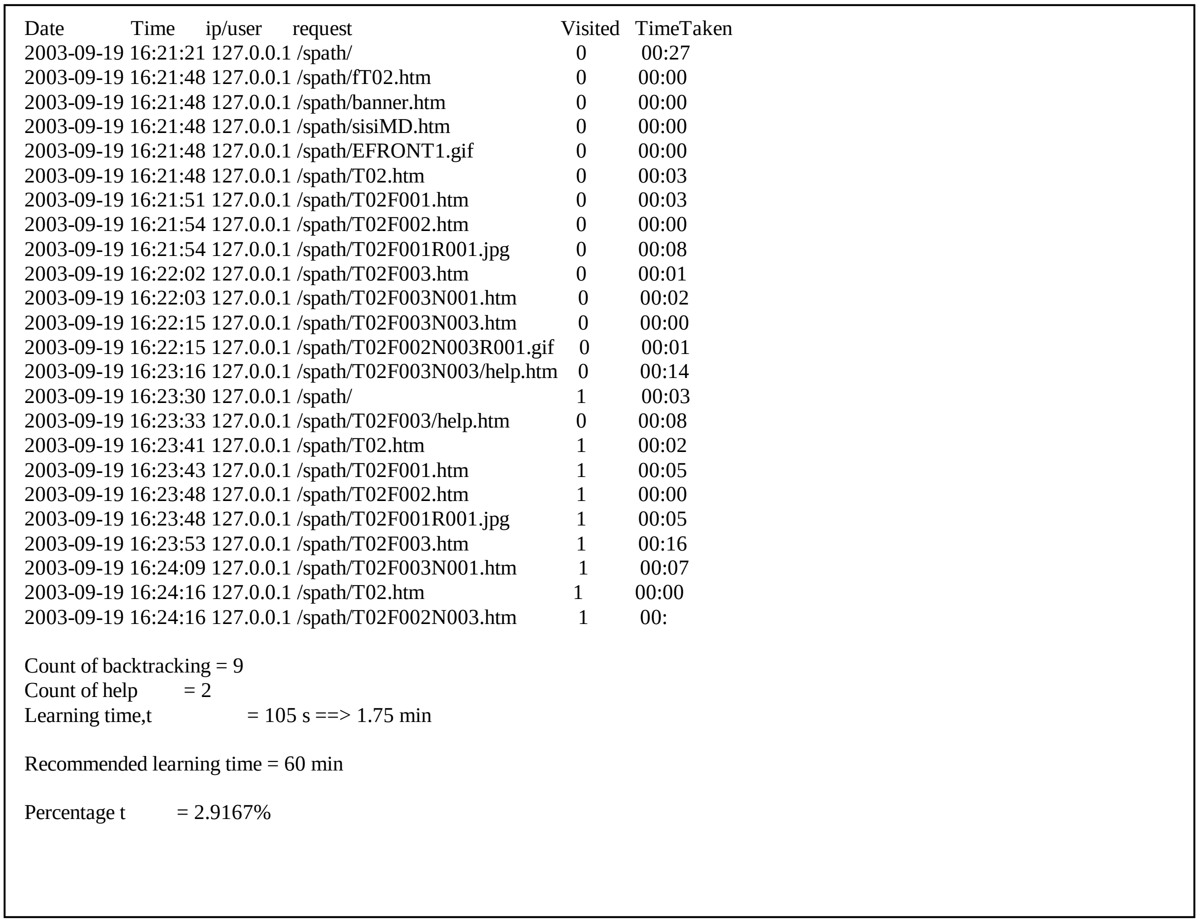
A session is defined by a unique IP address and a unique request time. It begins when user login or when system come across an IP address. The request time is defined as the beginning of a session. The system then keep tracking that individual's requests continuously, and define the end of that particular session for that individual when a subsequent request does not appear within an hour. Figure 4 shows a session extracted from the original web log file that has been analysed.

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2003-09-19 16:03:54
#Fields: date time c-ip cs-uri-stem cs(Referer)
2003-09-19 16:03:54 127.0.0.1 /iishelp/iis/htm/core/iiauths.htm
http://127.0.0.1/iishelp/iis/htm/core/iiabasc.htm
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2003-09-19 16:21:21
#Fields: date time c-ip cs-uri-stem
2003-09-19 16:21:21 127.0.0.1 /spath/
2003-09-19 16:21:48 127.0.0.1 /spath/fT02.htm
2003-09-19 16:21:48 127.0.0.1 /spath/banner.htm
2003-09-19 16:21:48 127.0.0.1 /spath/sisiMD.htm
2003-09-19 16:21:48 127.0.0.1 /spath/EFRONT1.gif
2003-09-19 16:21:48 127.0.0.1 /spath/T02.htm
2003-09-19 16:21:51 127.0.0.1 /spath/T02F001.htm
2003-09-19 16:21:54 127.0.0.1 /spath/T02F002.htm
2003-09-19 16:21:54 127.0.0.1 /spath/T02F001R001.jpg
2003-09-19 16:22:02 127.0.0.1 /spath/T02F003.htm
2003-09-19 16:22:03 127.0.0.1 /spath/T02F003N001.htm
2003-09-19 16:22:15 127.0.0.1 /spath/T02F003N003.htm
2003-09-19 16:22:15 127.0.0.1 /spath/T02F002N003R001.gif
2003-09-19 16:23:30 127.0.0.1 /spath/
2003-09-19 16:23:41 127.0.0.1 /spath/T02.htm
2003-09-19 16:23:43 127.0.0.1 /spath/T02F001.htm
2003-09-19 16:23:48 127.0.0.1 /spath/T02F002.htm
2003-09-19 16:23:48 127.0.0.1 /spath/T02F001R001.jpg
2003-09-19 16:23:53 127.0.0.1 /spath/T02F003.htm
2003-09-19 16:24:09 127.0.0.1 /spath/T02F003N001.htm
2003-09-19 16:24:16 127.0.0.1 /spath/T02F003N003.htm
2003-09-19 16:24:16 127.0.0.1 /spath/T02F002N003R001.gif
```

Figure 3: Example of Original Web Log Data

```
2003-09-19 16:21:21 127.0.0.1 /spath/
2003-09-19 16:21:48 127.0.0.1 /spath/fT02.htm
2003-09-19 16:21:48 127.0.0.1 /spath/banner.htm
2003-09-19 16:21:48 127.0.0.1 /spath/sisiMD.htm
2003-09-19 16:21:48 127.0.0.1 /spath/EFRONT1.gif
2003-09-19 16:21:48 127.0.0.1 /spath/T02.htm
2003-09-19 16:21:51 127.0.0.1 /spath/T02F001.htm
2003-09-19 16:21:54 127.0.0.1 /spath/T02F002.htm
2003-09-19 16:21:54 127.0.0.1 /spath/T02F001R001.jpg
2003-09-19 16:22:02 127.0.0.1 /spath/T02F003.htm
2003-09-19 16:22:03 127.0.0.1 /spath/T02F003N001.htm
2003-09-19 16:22:15 127.0.0.1 /spath/T02F003N003.htm
2003-09-19 16:22:15 127.0.0.1 /spath/T02F002N003R001.gif
2003-09-19 16:23:30 127.0.0.1 /spath/
2003-09-19 16:23:41 127.0.0.1 /spath/T02.htm
2003-09-19 16:23:43 127.0.0.1 /spath/T02F001.htm
2003-09-19 16:23:48 127.0.0.1 /spath/T02F002.htm
2003-09-19 16:23:48 127.0.0.1 /spath/T02F001R001.jpg
2003-09-19 16:23:53 127.0.0.1 /spath/T02F003.htm
2003-09-19 16:24:09 127.0.0.1 /spath/T02F003N001.htm
2003-09-19 16:24:16 127.0.0.1 /spath/T02F003N003.htm
2003-09-19 16:24:16 127.0.0.1 /spath/T02F002N003R001.gif
```

Figure 4: User Session 1



```
Date            Time      ip/user     request                                Visited   TimeTaken
2003-09-19 16:21:21 127.0.0.1 /spath/                                    0          00:27
2003-09-19 16:21:48 127.0.0.1 /spath/fT02.htm                            0         00:00
2003-09-19 16:21:48 127.0.0.1 /spath/banner.htm                          0         00:00
2003-09-19 16:21:48 127.0.0.1 /spath/sisiMD.htm                          0         00:00
2003-09-19 16:21:48 127.0.0.1 /spath/EFRONT1.gif                         0         00:00
2003-09-19 16:21:48 127.0.0.1 /spath/T02.htm                             0         00:03
2003-09-19 16:21:51 127.0.0.1 /spath/T02F001.htm                         0         00:03
2003-09-19 16:21:54 127.0.0.1 /spath/T02F002.htm                         0         00:00
2003-09-19 16:21:54 127.0.0.1 /spath/T02F001R001.jpg                     0         00:08
2003-09-19 16:22:02 127.0.0.1 /spath/T02F003.htm                         0         00:01
2003-09-19 16:22:03 127.0.0.1 /spath/T02F003N001.htm                     0          00:02
2003-09-19 16:22:15 127.0.0.1 /spath/T02F003N003.htm                     0          00:00
2003-09-19 16:22:15 127.0.0.1 /spath/T02F002N003R001.gif     0          00:01
2003-09-19 16:23:16 127.0.0.1 /spath/T02F003N003/help.htm    0          00:14
2003-09-19 16:23:30 127.0.0.1 /spath/                                    1          00:03
2003-09-19 16:23:33 127.0.0.1 /spath/T02F003/help.htm                0         00:08
2003-09-19 16:23:41 127.0.0.1 /spath/T02.htm                             1         00:02
2003-09-19 16:23:43 127.0.0.1 /spath/T02F001.htm                         1         00:05
2003-09-19 16:23:48 127.0.0.1 /spath/T02F002.htm                         1         00:00
2003-09-19 16:23:48 127.0.0.1 /spath/T02F001R001.jpg                     1         00:05
2003-09-19 16:23:53 127.0.0.1 /spath/T02F003.htm                         1         00:16
2003-09-19 16:24:09 127.0.0.1 /spath/T02F003N001.htm                     1          00:07
2003-09-19 16:24:16 127.0.0.1 /spath/T02.htm                            1         00:00
2003-09-19 16:24:16 127.0.0.1 /spath/T02F002N003.htm                     1          00:

Count of backtracking = 9
Count of help          = 2
Learning time,t                   = 105 s ==> 1.75 min

Recommended learning time = 60 min

Percentage t          = 2.9167%
```

Figure 5: The Example of Output of Program To Calculate Training Data For Classifier II

Data collected from web log file include date of request, time of request, IP address/username and page request. In this phase, we need to train the data before using it with neural network to perform student classification task. One more program is developed to calculate the desired output. Input for this program will be time and request. Figure 5 is the example of output of this program. The recommended learning time is 60 minutes.

Table 2 shows a few criteria identified to be an input data for the second classifier. The criteria include the student's learning time, number of backtracking, and number of using help (Hashim et al., 2001). We identified the student's status by comparing the learning time spent by the student with the time recommended by system. If the student took more than the recommended time in learning, this indicates that he is a slow learner. On the other hand, if he took less time, he is a fast learner.

In this research, we assumed that a beginner student will backtrack five times to the relevant materials he has gone through earlier as shown in Table 2. If the student used the help facilities frequently, it showed that the student is having problem understanding the learning module. Therefore, the number of using help facilities was considered as one of the criteria to classify students.

Table 2: Criteria to Classification II

Criteria	Beginner, α	Intermediate, β	Advanced, γ
Learning Time, x_1	$x_1 > 100\%$	$80\% \le x_1 \le 100\%$	$x_1 < 80\%$
Freq. Of Backtracking, x_2	$x_2 > 5$	$3 \le x_2 \le 5$	$x_2 \le 1$
Freq. Of Using Help, x_3	$x_3 > 5$	$3 \le x_3 \le 5$	$x_3 \le 1$

Mathematically,

$$\alpha(x_i) \in \left\{ \begin{matrix} x_1 > 100\% \\ x_2 > 5 \\ x_3 > 5 \end{matrix} \right\} \quad \beta(x_i) \in \left\{ \begin{matrix} 80\% \le x_1 \le 100\% \\ 3 \le x_2 \le 5 \\ 3 \le x_2 \le 5 \end{matrix} \right\} \quad \gamma(x_i) \in \left\{ \begin{matrix} x_1 < 80\% \\ x_2 \le 1 \\ x_3 \le 1 \end{matrix} \right\}$$

Data Representation:

Beginner	: 00
Intermediate	: 01
Advanced	: 11

5.0 INTEGRATION OF EXPLICIT AND IMPLICIT TECHNIQUES: CLASSIFIER III

For the third classifier, we integrated the explicit and implicit data extraction to obtain the input data for the student classification.

5.1 Training Data Sample

Students' behavior data are collected from analysis of web log data similar to the second classification process. The data considered are the requested pages and time of request. In the explicit user data extraction process, we collect students' score value from the system's database as an additional input data into our program to calculate the desired output data for the training sample. The question and answer session is used to test the student's understanding of the material being learned. Table 3 shows the possible criteria for input data to calculate the desired output.

Table 3: Criteria To Classification III

Criteria	Beginner, α	Intermediate, β	Advanced, γ
Learning Time, x_1	$x_1 > 100\%$	$80\% \leq x_1 \leq 100\%$	$x_1 < 80\%$
Freq. Of Backtracking, x_2	$x_2 > 5$	$3 \leq x_2 \leq 5$	$x_2 \leq 1$
Freq. Of Using Help, x_3	$x_3 > 5$	$3 \leq x_3 \leq 5$	$x_3 \leq 1$
Score, x_4	$x_4 < 60\%$	$60\% \leq x_4 \leq 800\%$	$x_4 > 80\%$

Mathematically,

$$\alpha(x_i) \in \begin{Bmatrix} x_1 > 100\% \\ x_2 > 5 \\ x_3 > 5 \\ x_4 < 60\% \end{Bmatrix} \quad \beta(x_i) \in \begin{Bmatrix} 80\% \leq x_1 \leq 100\% \\ 3 \leq x_2 \leq 5 \\ 3 \leq x_2 \leq 5 \\ 60\% \leq x_4 \leq 800\% \end{Bmatrix} \quad \gamma(x_i) \in \begin{Bmatrix} x_1 < 80\% \\ x_2 \leq 1 \\ x_3 \leq 1 \\ x_4 > 80\% \end{Bmatrix}$$

Data Representation:

Beginner	: 00
Intermediate	: 01
Advanced	: 11

6.0 CONCLUSION AND FUTURE WORK

Data preprocessing is implemented through the process of extracting information from registration data (explicit technique) and web log analysis (implicit technique) to develop a complete student profile for the student who use the system. We defined three types of classifiers to be used in classifying the students' status based on input data collected from the registration form, web log data or combination of the web log data and the students' score while doing the exercises.

The results from this phase is a set of training data including input data and desired output data that will be used in classification with neural network for future work. The interesting issue here concerned with the session identification using student's metric number. In this paper we consider IP address as a unique session identifier. For future research, we will look at the integration of metric number variable, IP address and the time of request to identify unique session.

Acknowledgements

This research is conducted under IRPA grant, vote project 74129: The Development of Knowledge Management For Adaptive Hypermedia Learning System, Faculty of Computer Science, UTM Skudai.

7.0 REFERENCES

1. Bailey, D. And Thompson, D. (1990). *How To Develop Neural-Network Applications*. In AI Expert. June 1990. 38-45.

2. Brusilovsky, P. (1999). *Adaptive and Intelligent Technologies For Web-Based Education*. In C. Rollinger and C. Peylo (eds) Kunstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching. 4. 19-25.

3. Buser, D., Kauffman, J., Llibre, J.T., Francis, B., Sussman, D., Ullman C. And Duckett J. (2000). *Application, Sessions And Cookies*. Wrox Press Ltd. (2000). Beginning Active Server Pages 3.0. 289-330.

4. Catledge, L.D. and Pitkow, J.E. (1995). *Characterizing Browsing Strategies in The World Wide Web*. Georgia Institute of Technology, Atlanta, USA.

5. Cooley, R., Mobasher, B. And Srivastava, J. (1997). Web Mining: Information and Pattern Discovery on

The World Wide Web. Department of Computer Science, University of Minnesota, USA.

6. Cripps, A. (1996). *Using Artificial Neural Nets to Predict Academic Performances*. In Association of Computing Machinery, Inc.(ACM). 33-37.

7. Hashim, S.Z.M., Yusof, N., Samsuri, P. and Ahmad, N.B. (2001). *Design and Implementation of An Adaptive Web-Based Hypermedia Learning System: A Prototype*. In Proceedings of The International Conference on Organizational Development and Leadership in Education. Kuala Lumpur, Malaysia.

8. Jeon-Slaughter, H. (2002). *Report of Web Log Data Mining*. In E-Journal User Study: Recent Findings. Stanford University Libraries.
URL: http://ejust.stanford.edu/logdata.pdf

9. Moore, D.J., Hobbs, D.J., Mullier, D. And Bell, C. (1997). *Interaction Paradigms With Educational Hypermedia*. In In Proceedings of 23rd EUROMICRO Conference'97 New Frontiers of Information Technology. Budapest, Hungary.

10. Mullier, D.J. (2000). *The Application of Neural Network And Fuzzy Logic Techniques To Educational Hypermedia*. Ph.D Thesis. Leeds Metropolitan University, UK.

11. Mullier, D.J. (1995). *Using A Neural Network To Model Hypermedia Browsing – An Alternative To Traditional Intelligent Tutoring Methods*. Faculty of Information and Engineering Systems, Leeds Metropolitan University, UK.

12. Quentin-Baxter, M. (1993). *Hypermedia Learning Environments Limit Access To Information*. Faculty of Medicine Computing Centre, University of Newcastle Upon Tyne, UK.
URL:
http://decweb.ethz.ch/WWW7/1941/com1941.htm

13. Wang, Y. (2000). *Web Mining and Knowledge Discovery of Usage Patterns*. In CS 748T Project Report (Part I).
URL:
http://db.uwaterloo.ca/~tozsu/courses/cs748t/surveys/wang.pdf

14. Yusof, N., Salim, A., Samsuri, P., Hashim, S.Z.M. and Ahmad, N.B. (2000). *SPAtH: An Adaptive Web-Based Learning System*. In Proceedings of IASTED International Conference. Banff, Canada.

15. Zaiane, O.R. and Luo, J. (2001). *Towards Learners' Behaviour in Web-Based Distance Learning Environment*. Department of Computing Science, University of Alberta, Canada.