Neural Comput & Applic (2014) 24:675–684
DOI 10.1007/s00521-012-1284-8

ORIGINAL ARTICLE

Determining students' level of page viewing in intelligent tutorial systems with artificial neural network

Abdulkadir Karacı · Nursal Arıcı

Received: 13 June 2012 / Accepted: 17 November 2012 / Published online: 29 November 2012

Abstract The concept of level of page viewing (LPV) refers to the extent to which a student actively revises the pages that he or she has to study in tutorial systems. In the present study, an artificial neural network (ANN) model, which is composed of 5 inputs, 20 and 30 neurons, 2 hidden layers, and 1 output, was designed to determine the students' LPV. After this network was trained, it was integrated into a web-based prototype teaching system, which was developed by ASP.net C# programming language. Additionally, Decision Tree method is tried to determine students' LPV. However, this method gave wrong results according to expected LPV values. In this system, the student first studies the pages uploaded by the teacher onto the system. After studying all the pages within the scope of a topic, the student can go to the test page for evaluation purposes. LPVs of a student who wants to navigate to the test page are calculated by an ANN module added to the system. On the condition that one or more of the LPV's are not up to the desired level, the student is not allowed to take the test and is informed of the pages with missing LPV's so that he can re-study these pages. This prototype system developed based on ANN to determine students' LPV is essential for intelligent tutorial systems, geared to provide intelligent assistance and guidance. The system can track the pages which the students did not study sufficiently and thus direct them to relevant pages. How much activity the students perform on each page to study is observed before they actually take the test, and the areas which should be further revised are determined much in advance.

Keywords Artificial neural network · Intelligent tutorial system · Level of page viewing · Learning management system

A. Karacı (✉)
Faculty of Education, Kastamonu University,
Kastamonu, Turkey
e-mail: akaraci@gmail.com

N. Arıcı
Faculty of Technology, Gazi University,
Beşevler, Ankara, Turkey
e-mail: nursal@gazi.edu.tr

1 Introduction

Artificial neural networks (ANNs) are structures that are inspired by neurobiological studies of the human brain [1]. An artificial neural network (ANN) does not need such a specific equation form. An ANN is a model composed of several highly interconnected computational units called neurons or nodes. The artificial neurons have input connections (dendrites), output connections (axons), and an internal process that generates an output signal in response to the input signal. Each node performs a simple operation on its inputs to generate an output that is forwarded to the next node in the sequence. This parallel processing brings great advantages for data analysis. Several studies have focused on ANN to deal with the problems involving incomplete or imprecise information [2–4].

In general, ANN are most useful in tasks such as model selection and classification, function estimate, determination of the optimum value, and data categorization [5]. The present study uses the function estimate aspect of the ANN.

The purpose of the computer-assisted education is to help students learn more effectively. To this end, the educational technologies have advanced rapidly, which ultimately have produced more effective teaching techniques. One of these techniques, online learning with intelligent tutorial system (ITS), has become quite popular [6, 7]. ITS is intelligent because it uses the methods and principles of artificial intelligence such as describing the knowledge level, inference mechanism, and machine learning [8].

ITS's are increasingly used in education because they provide the students with an environment conducive to independent and personalized learning, give them intelligent assistance and guidance, enable them to study independently of time and space [6].

An ITS is an invaluable tool that helps students learn in and out of class by means of educational materials. It enhances learning by personalizing the learning environment [9].

There is considerable evidence in the literature that most ITS's allow students to go to the test page without considering how much they studied and viewed the study pages [7, 10–17]. This means students are evaluated, although they take the test without studying the relevant pages sufficiently. A student who takes the test without revising the related pages sufficiently fails the test and thus loses motivation.

The present study proposes an ANN-based method to identify students' levels of page viewing and develops a prototype web-based teaching system wherein this method is used. The concept of level of page viewing (LPV) refers to the extent to which a student actively studies the pages that he or she is responsible for. Although in the literature, the number and time of page viewing are available, the level of page viewing is not included. This term, within the scope of a developed ANN model, is used to express at what level the students work on the page according to the number and the time of page views. The student first revises the pages uploaded by the teacher onto the system. After revising all the pages related to the subject, the student can go to the test page for evaluation purposes. LPVs of a student who wants to go to the test page is calculated by the ANN module added to the system. If one or more of the LPV's are not at the required level, the student is not let to take the test, is informed of the pages marked as LPV missing, and is required to study these pages once again. Thus, the student cannot take the test unless he or she studies the pages sufficiently. The details concerning the calculation of LPV by ANN are given in the methodology section.

2 Related works

Several studies in the literature have developed ITSs by using different methods. Rishi et al. proposed an ITS model to help students learn interactively and at their own pace in a student-centered environment. The model proposed maximizes the interaction between the students and ITS, and personalizes the learning process for the individual needs of students [7]. BITS [10] was developed as an ITS teaching computer programming (C++). Sheng-Jen HSIEH and Patricia Yee HSIEH used "XAIDA," the ITS design tool, and developed an intelligent tutorial system for the teaching of CNC machines [11]. ZOSMAT [12] is an ITS developed for math education. The student can use this system by himself as well as in the classroom under the supervision of an expert trainer. The system was piloted in a group of totally 80 students, and a follow-up test was given. The results indicated that the students who used this system scored an average of 90, whereas those who did not use the system scored an average of 66. SQL-Tutor is an ITS designed to teach the SQL language [13]. CAPIT [14] was developed as an ITS using constraint-based modeling for punctuation marks and capitalization. It is a problem-based system. Pl@tos [15] is an ITS, which can compile, interpret, and teach the script programming languages. All the subjects are initially marked as *not-learned* for all students in this system. It is decided whether the student has comprehended the subject based on the scores of a test implemented afterward. MathITS [16] is an ITS estimating the difficulty level of the items for math education by using a differential equation. In the system developed, based on the students' responses to certain questions, the probability of their accurately responding to other related questions is estimated. The estimated probability is at the same time indicative of students' achievement of the subject. Acampora et al. propose a new computational intelligence method for optimal personalized learning activity. This method proposed the use of the memetic optimization theory embedded in a hierarchical distribution scenario to derive the fittest matching between learning paths and available learning activities in order to generate the best personalized e-Learning experiences [17].

3 Significance of the study

In this study, a prototype web-based teaching system was developed using modern software technologies such as ASP.net C# programming language, Ajax and Jquery. Within this prototype developed, an ANN with 5 inputs, 1 output, 20 and 30 neurons, and 2 hidden layers was developed to determine students' LPV. The network was trained by presenting some of the data previously established to the network as the training set. Upon completion of the training of the network, data that had never been entered to the network before the training was entered to the network as input. When the actual output and the expected output were compared, it was seen that the network output produced results that are at the desired level of accuracy. Therefore, the ANN was integrated into the prototype teaching system developed. The prototype education system developed calculates students' LPV by ANN and directs them to the pages with insufficient LPV's. The students are made to revise all the pages properly before they take the test, thereof. This prototype system developed based on ANN to determine the students' LPV is crucial for ITS's which are to provide intelligent assistance and guidance.

4 Method

Most significantly, in the prototype teaching system developed, a content management system which enables teachers to upload lecture content was established. The teacher uploads the necessary pages related to the subject onto the system by the help of the content management system, and the student logs in to study the pages uploaded by the teacher. As the students study the pages, the prototype education system continuously follows them, recording into the database their page entry/exit dates and hours, the lengths of study. When a student wants to go to the test page, the prototype education system calculates the student's LPV by the help of ANN integrated into the system. If the LPV calculated is not up to the level desired by the teacher, the student is asked to re-study these pages. For this calculation, ANN uses the frequency and duration of visits made by students to the page. To enter the test page for the student to reach, the LPV is determined by the teacher in the form of linguistic expression "Good," "Fair," and "Poor" as in Fig. 2. The limit values of these linguistic expression criteria are stated under the heading of "output layer;" for instance, if the LPV criterion to which a student is required to reach is selected as "Fair" for any page, when the student's LPV is "Good" or "Fair," desired level is thought to be reached.

Teacher should determine the page viewing time boundary values and LPV to be reached for every page according to students those have higher level of understanding skills. This situation is resulted that lower level students may take examinations even though they studied the pages less than required time. But, developed prototype ITS and other ITS identify the lower succeeded pages and lead students to those pages for re-study. For this re-study session, the system does not take into account of previous page viewing time and frequency. Therefore, lower level students have to study the pages more than higher level of students. For higher level of students, it takes less time to move on to next topic.

4.1 The structure and the parameters of the artificial neural network developed

The learning performance in the training and test stages was assessed based on mean squared error (MSE) and R (regression value) coefficients in this study. If the MSE coefficient is close to zero, then there is little difference between the actual output and the desired output. If MSE is zero, then there is no difference between the output of the network and the expected output, which means there is no fault. R, on the other hand, determines the strength of the relationship between the network output and the expected output. If this value is 1, the relationship between the network output and the expected output is not coincidental [1]. There are lot of methods to classify the data. In this study, ANN and Decision Tree methods are tried for classification. However, ANN method is preferred. Because of R and MSE parameters shown at Table 4 in Sect. 5, ANN performs the process of classification for training and test data at the desired error level. Additionally, ANN method performs better classification than Decision Tree method. Moreover, ANN has high tolerance for training of noisy data [18]. ANN is said to learn the rules from the examples. In contrast, a traditional rule-based system would have rules encoded within it that a designer has previously identified [19].

An ANN using back-propagation algorithm as the teaching algorithm and comprised of 5 inputs, 1 output, and 2 hidden layers was used in the system which is developed to identify students' LPV. The number of neurons in the hidden layer was chosen to be 20 and 30. The ANN model used in the system developed is shown in Fig. 1.

Input Layer: Comprised of 5 neurons. The information in Table 1 is given as input to these neurons.

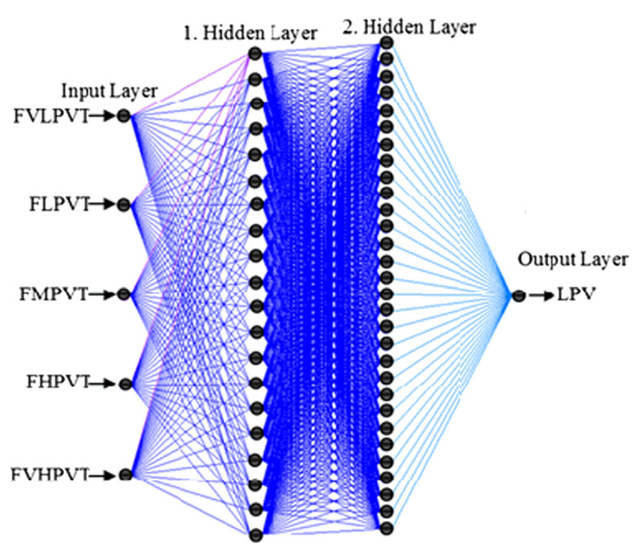

Fig. 1 The ANN model used in the system developed

The inputs in the prototype teaching system are calculated based on users' entry/exit hour, date, and time spent on page as recorded for each page. Frequency of page viewing is operationalized according to 5-category length boundary values defined along "Very Low," "Low," "Moderate," "High," and "Very High." The boundary values of the page viewing time "Very Low," "Low," "Moderate," "High," and "Very High" are determined for each page by the teacher on a range between lower boundary and upper boundary. By this way, the duration of stay required for a page is specially defined for each page, and the system calculates the LPV of each page according to this definition. This allows for customization according to the difficulty level of the page, making it possible to define the length of study required for each page independently from other pages. The screen, on which there are the boundary definitions of the page viewing time for each page according to 5 categories, is shown in Fig. 2.

Hidden Layer: ANN designed has two hidden layers. The number of neurons in the hidden layer is a parameter influencing the learning performance. If the number of the neurons in the hidden layer is too small, then the network earning procedure cannot converge to an optimal value, showing an oscillatory behavior of the error function. Therefore, the network cannot learn the relationship between the input–output designs. If the number of the neurons is too high, then the network will just store the input–output list, showing a poor generalization performance. This means that the optimal ANN size should fit the data structure and construct the model definitively related to the problem [20]. As a result of trials, the number of neurons in the hidden layer for the first ANN was found to be 20, while it was found to be 30 for the second ANN. It is because the optimum result was obtained when the number of neurons in the hidden layer had such values.

Output Layer: ANN is composed of 1 neuron output and gives the LPV value numerically as output. The LPV value, which is the output of ANN, is converted into the linguistic expressions of "poor," "fair," and "good" using the boundary values in Table 2. This conversion process takes place within the prototype teaching system. Moreover, the output of training data used to train the network was established based on Table 2.

4.2 Training of the artificial neural network

MATLAB program was used to design and train the artificial neural network in this study. The testing and training procedures were conducted by presenting the frequency of very low, low, moderate, high, and very high page viewing time to the ANN model as input and taking the LPV value as output. In the ANN model, a total of 132 data were used for training and 30 data which the network had never encountered during the training were used for the test

Table 1 Artificial neural network inputs

Input	Explanation
FVLPVT	Frequency of very low page viewing time
FLPVT	Frequency of low page viewing time
FMPVT	Frequency of moderate page viewing time
FHPVT	Frequency of high page viewing time
FVHPVT	Frequency of very high page viewing time

Table 2 The scale range used to convert ANN output to LPV verbal expressions

ANN output (LPV quantitative value) boundary values	LPV verbal expressions
$0.5 \leq LPV < 1.5$	Poor
$1.5 \leq LPV < 2.5$	Fair
$2.5 \leq LPV < 3.5$	Good

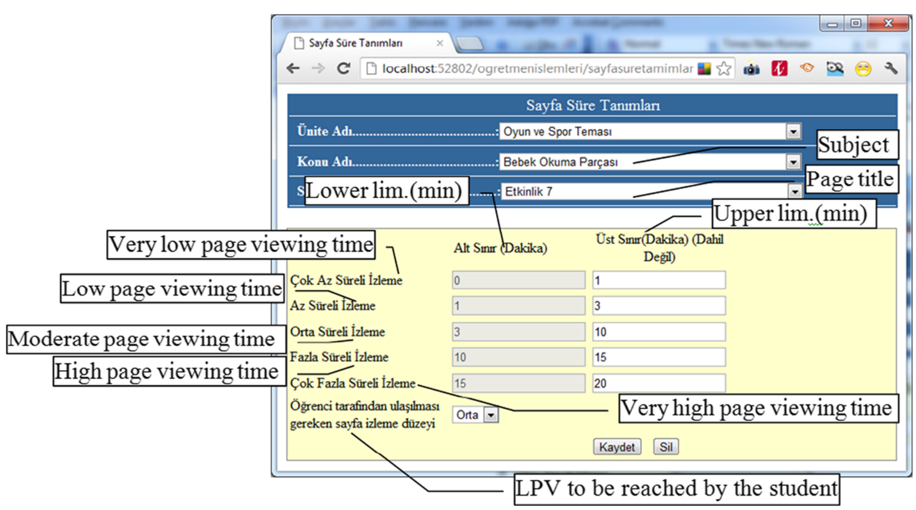

Fig. 2 The screen displaying the boundary values of page viewing time

Table 3 The values of parameters used in the model

Parameters	ANN
Number of input layer neurons	5
Number of hidden layer	2
Number of first hidden layer neurons	20
Number of second hidden layer neurons	30
Number of output layer neuron	1
Learning cycle	179 Epochs
Transfer function	Tansig
Training function	TRAINLM
Performance function	MSEREG

procedure of the ANN modeling. The data used for training and test were compiled by the researchers and also by consulting expert opinion. According to ANN model which is generated by taking into account the number and time of the student's page views, the training and test data in experimental or depending on a particular rule creation are not possible. Therefore, these data must be established by domain experts. This situation is similar to creating the base rules in fuzzy logic. What is important here is the model created in order to determine the level of page viewing. Different training and test data which are suitable for the input and output parameters of this model can be created by other field experts. According to the data generated, re-trained ANN can be used to determine the level of page viewing. Emphasized strongly here is to provide a model by defining input and output parameters.

The values of parameters used in ANN are given in Table 3.

For the ANN, which is trained in Matlab, to be used in ASP.net environment, the program codes were generated in Matlab environment as .m file. The codes generated were transformed into "dll" file by using Matlab Deployment Tool for .Net. Dll file was defined as library to be used in prototype teaching system developed by C# language in asp.net environment.

4.3 Structure of decision tree

A decision tree is a graphical representation of a procedure for classifying or evaluating an alternative of interest. By graphical representation, they clearly show how to reach a decision and they are able to construct automatically from labeled instances. Generally speaking, the basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top–down recursive divide-and-conquer manner [21, 22]. Decision tree constructed for this paper is trained by same data set that is used to train ANN. Training data consisted of 132 values. It also tested with

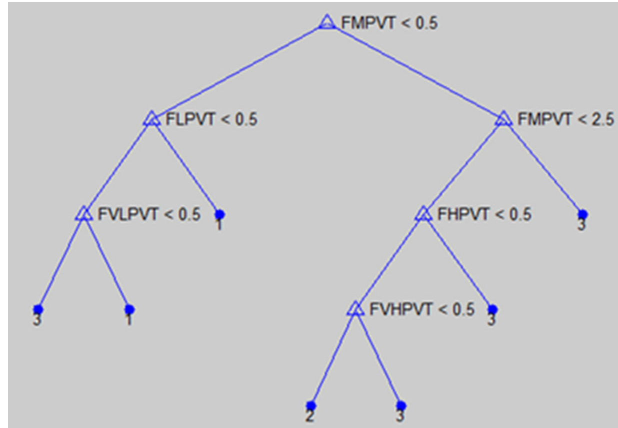

Fig. 3 Constructed decision tree

same 30 value data set which is used to test ANN. Constructed decision tree is shown at Fig. 3.

Classification rules for decision tree at Fig. 3 are as follows:

1. if FMPVT $<$ 0.5 then node 2 else node 3
2. if FLPVT $<$ 0.5 then node 4 else node 5
3. if FMPVT $<$ 2.5 then node 6 else node 7
4. if FVLPVT $<$ 0.5 then node 8 else node 9
5. fit = 1
6. if FHPVT $<$ 0.5 then node 10 else node 11
7. fit = 3
8. fit = 3
9. fit = 1
10. if FVHPVT $<$ 0.5 then node 12 else node 13
11. fit = 3
12. fit = 2
13. fit = 3

5 Results and discussion

The learning performance in the training and test stages was assessed based on MSE and regression value (R) coefficients in this study. The MSE and R values calculated for the training and testing sets after the training of the ANN and decision tree are displayed in Table 4.

REG and MSE values in the ANN model was found to be 0.9998 and 3.0461e−004 for the training set and 0.9991 and 9.6042e−004 for the testing set. It was observed that the R values are very close to 1 for both the training set and the testing set. This indicated that the relationship between the actual output and the desired output is not coincidental. MSE coefficients also appear to be very close to 0 for both the training set and the testing set. As these values are highly close to zero, there is little, if any, difference between the network output and the desired output. To sum up, the R and MSE values point to the fact that the network designed in the ANN model was effectively trained.

REG and MSE values in the Decision Tree method was found to be 1 and 0 for the training set and 0.6997 and 0.4 for the testing set. R and MSE parameters for training set may be considered at perfect level. MSE parameter of decision tree is larger than ANN method while R parameter of decision tree is smaller than ANN method for test data. Important point here is giving the correct results to test data that is not used during training. When we look at results 23, 26, and 29 at Table 5, decision tree gave completely wrong result compared to expected outcome. For these reasons, ANN method is preferred for classification.

Table 4 The statistical values of the proposed ANN and Decision Tree methods

Statistical parameters	ANN		Decision tree	
	Training set	Testing set	Training set	Testing set
REG (R)	0.9998	0.9991	1	0.6997
MSE	3.0461e−004	9.6042e−004	0	0.4

The regression analysis graphs of R parameters for the training and testing set of ANN and Decision Tree methods are given in Fig. 4.

The results obtained by entering the test data, which is not used in the training of the network and decision tree, into the trained network and decision tree, and the expected results concerning these data are shown in Table 5.

Table 5 demonstrates that when the test data were given to the network, the expected network output turned out to

Table 5 Comparison of expected outcomes and the estimated outcomes obtained from ANN and decision tree for the test set

Order	Input					Output (student's LPV values and ling. expression)					
	FVLPVT	FLPVT	FMPVT	FHPVT	FVHPVT	Expected output	Ling. expr.	ANN	Ling. expr.	Decision tree	Ling. expr.
1	5	5	2	0	0	2	Fair	2.02	Fair	2	Fair
2	1	1	1	1	1	3	Good	2.96	Good	3	Good
3	2	2	2	0	0	2	Fair	2.03	Fair	2	Fair
4	3	3	3	0	0	3	Good	2.95	Good	3	Good
5	5	5	10	0	0	3	Good	3	Good	3	Good
6	0	0	15	1	0	3	Good	3	Good	3	Good
7	4	4	12	4	0	3	Good	3	Good	3	Good
8	7	7	3	1	1	3	Good	3	Good	3	Good
9	3	5	1	0	0	2	Fair	1.97	Fair	2	Fair
10	15	15	1	0	1	3	Good	2.99	Good	3	Good
11	6	3	4	5	0	3	Good	3	Good	3	Good
12	100	0	0	0	0	1	Poor	1.01	Poor	1	Poor
13	0	100	0	0	0	1	Poor	1	Poor	1	Poor
14	1	1	1	1	15	3	Good	3	Good	3	Good
15	1	0	5	1	0	3	Good	3	Good	3	Good
16	5	5	1	0	0	2	Fair	1.97	Fair	2	Fair
17	1	1	2	2	0	3	Good	3	Good	3	Good
18	10	30	3	0	0	3	Good	2.94	Good	3	Good
19	20	20	1	0	0	2	Fair	1.93	Fair	2	Fair
20	0	0	2	0	0	2	Fair	2.03	Fair	2	Fair
21	128	3	2	0	0	2	Fair	2.06	Fair	2	Fair
22	110	0	0	0	0	1	Poor	1.01	Poor	1	Poor
23	90	90	0	1	5	3	Good	3	Good	1	Poor
24	49	0	0	0	0	1	Poor	1.01	Poor	1	Poor
25	196	1	3	1	0	3	Good	3	Good	3	Good
26	107	1	0	0	2	3	Good	3	Good	1	Poor
27	131	1	3	0	1	3	Good	3	Good	3	Good
28	116	0	1	0	0	2	Fair	1.92	Fair	2	Fair
29	116	1	0	3	0	3	Good	3	Good	1	Poor
30	267	2	2	0	3	3	Good	3	Good	3	Good

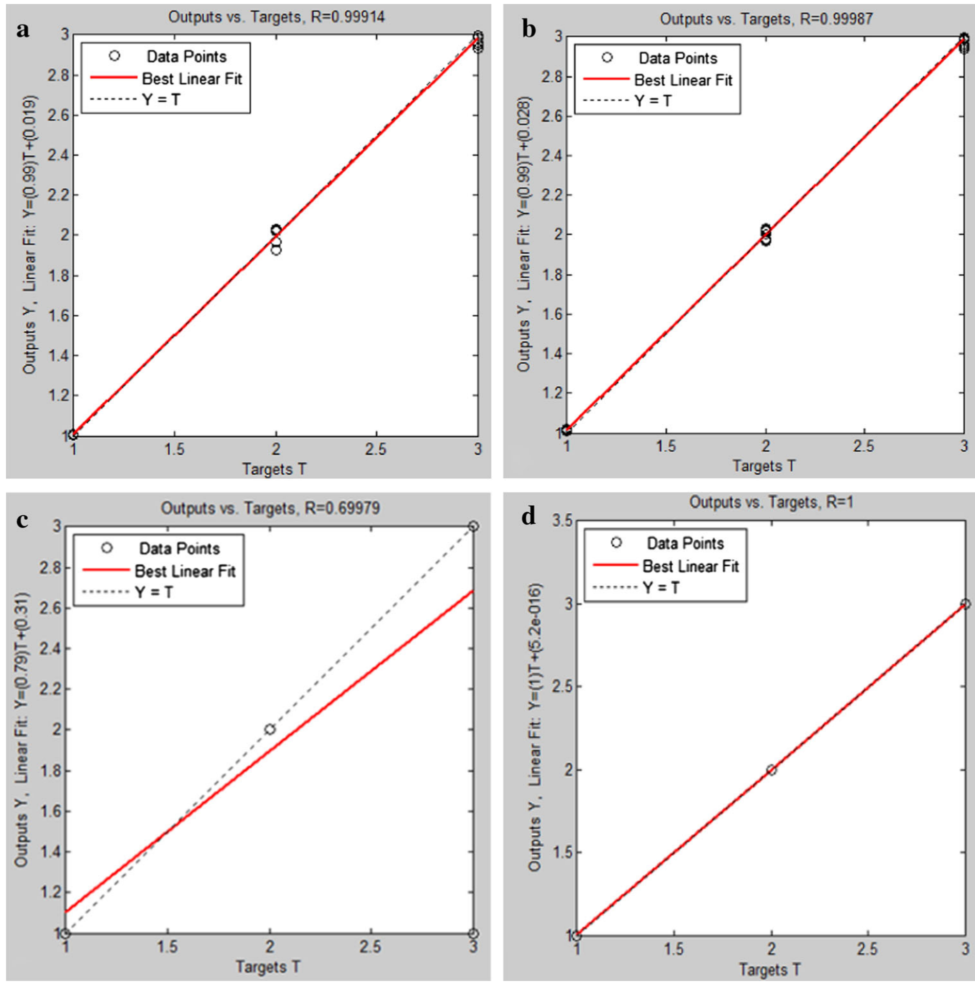

Fig. 4 **a** The regression graph of ANN model testing set. **b** The regression graph of ANN model training set. **c** The regression graph of Decision Tree method testing set. **d** The regression graph of Decision Tree method training set

be very close to the actual output. It was also observed that, when the network output was converted to linguistic expression and compared, the expected linguistic expression and the linguistic expression obtained from the network are truly identical; for instance, in the test data number 1, the "Frequency of Moderate Page Viewing Time (FMPVT)" was 2, the output of the network as student's LPV linguistic expression was calculated to be "fair." It matches the student's LPV linguistic expression expected at this level. Similarly, in the test data number 5, the FMPVT was 10, the output of the network as student's LPV linguistic expression was calculated to be good. As the "frequency of moderate page viewing time" increased, student's LPV linguistic expression rose to the next high level. The fact that the data set which is to be used for the training of the network was designed for such situations accounts for this output of the network.

The system underwent some trials after the page contents were uploaded to the prototype teaching system, and ANN was incorporated into the model. As can be seen in Fig. 5, when the student studies the pages and wants to take the follow-up test, the pages which do not bear "good" LPV are reported to the students. In this screen, the student can click on the "Go to Page" link to revisit and revise the pages with low LPV. The student is not entitled to take the test until he or she sufficiently studies these pages and reaches the level "good."

As can be seen in Fig. 5, the pages with insufficient student's LPV are reported to students; the teacher also can see any student's LPV by using the "Student Monitoring" module. Figure 6 demonstrates a sample screen used by the teacher to list a student's LPV. This screen details the LPV's concerning all the pages the student has studied. Moreover, the teacher is separately notified of statistics relating to pages with LPV linguistic expressions "Bad," "Fair," and "Good." By this way, the teacher can trace which student has studied which page and for how long.

If the teacher wants to reach more detailed information about the date and hour the student entered/exited the page and how long the visit lasted, he or she can obtain this

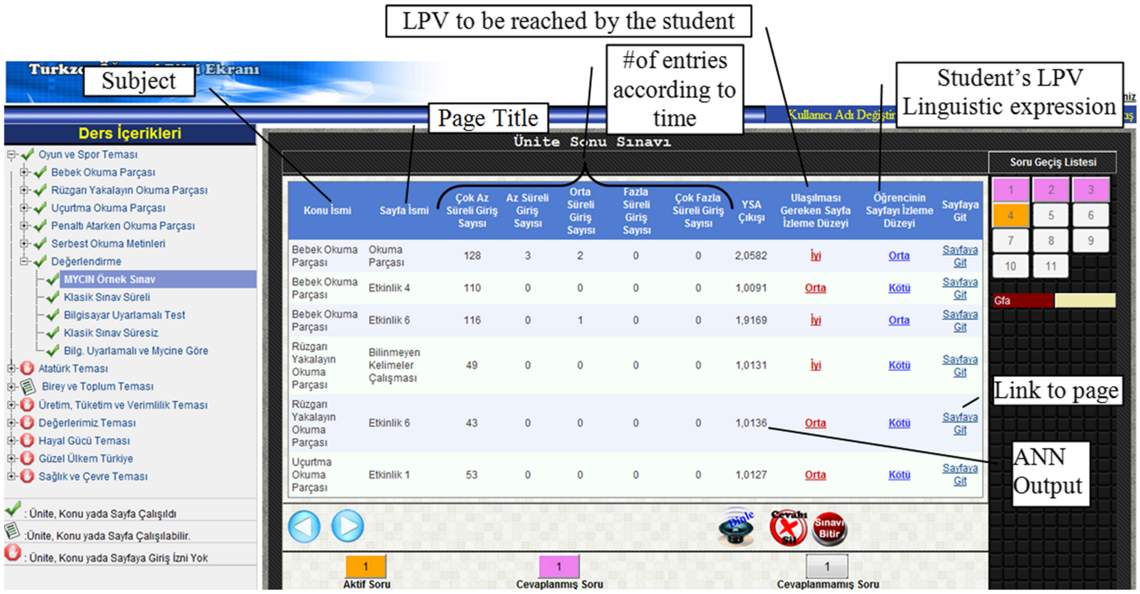

Fig. 5 Testing the student interface of prototype teaching system into which ANN was integrated

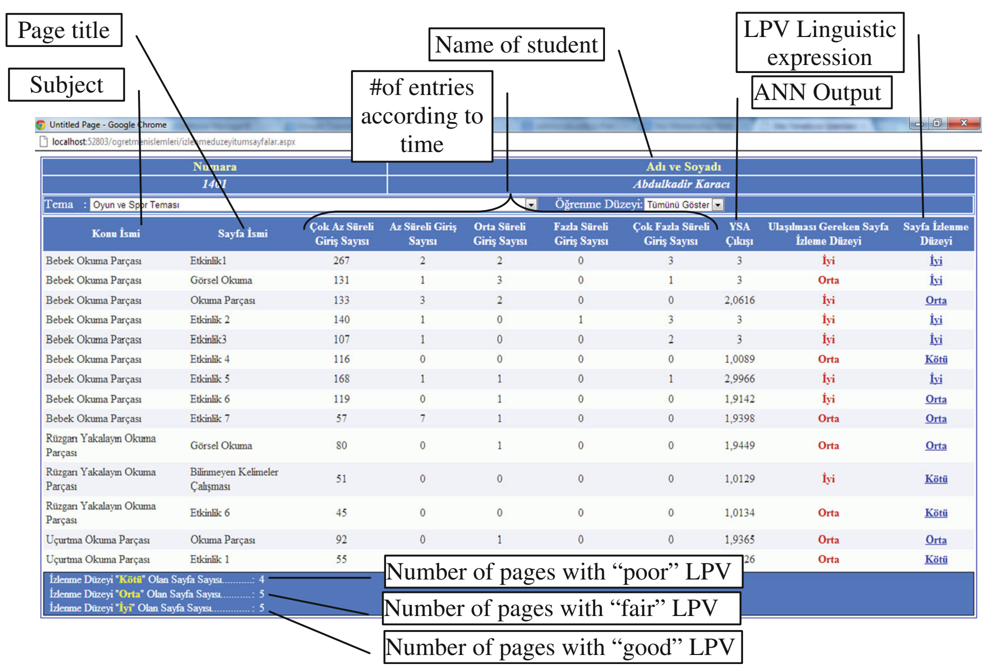

Fig. 6 A sample list of LPV's made by teacher

information from a screen such as the one in Fig. 7. For example, the screen in Fig. 7 reveals for the teacher the following information on a student named "Abdulkadir Karacı:" that he studied, the entry/exit hours and dates for the page "Okuma Parçası" ("The Reading Passage"), total time spent on this page, frequency of page viewing according to time spent on page (frequency of low page viewing time, frequency of moderate page viewing time, etc.), the total number of visits to this page, total number and average of minutes spent on the page, and LPV linguistic expression. In this page, different from that in Fig. 6, each page a student has studied can be analyzed in detail.

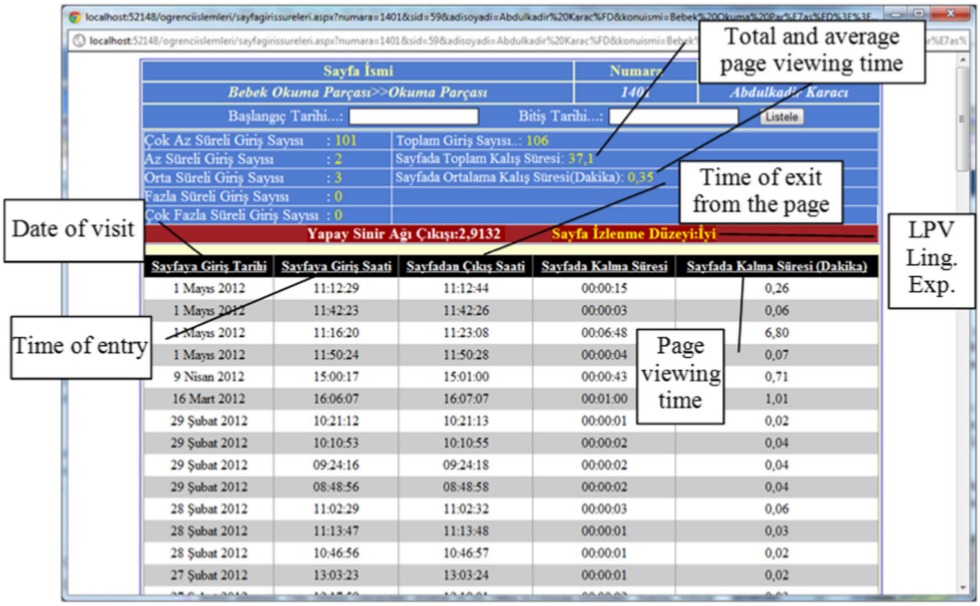

Fig. 7 Details about the page viewing time

6 Conclusions

In the present study, the prototype teaching system developed reveals the LPV, which indicates how actively students study the pages, in the form of linguistic expressions based on ANN. This prototype system developed based on ANN to identify students' LPV's is essential for intelligent tutorial systems, which have to provide intelligent assistance and guidance. The system can identify the pages insufficiently studied and thus direct the students to the relevant pages. By this way, the students are closely monitored before they take the test, and the pages which need further revision are identified well in advance. In most learning management systems (LMS), log files are kept to track the extent to which the students view the web pages; these files keep record of the date and hour of visits to the web pages. However, it is an extremely demanding procedure for teachers and trainers to analyze such complex data in the log files for each and every student to identify their levels of page viewing. The system we have developed evaluates these data by means of ANN instantly and with only a few mouse clicks and effectively reports LPV in the form of linguistic expression.

In conclusion, this system used to identify LPV is suitable to both guide students in ITS's and inform the teachers in education management systems. In brief, estimation of LPV by ANN is a practical and effective method.

7 Future study

In this study, students who study the pages less than desired level blocked from taking the examination and forced to re-study those pages. It is obvious that this method is both efficient and necessary than conducting the examination without any control. In next study, recommended method of this study will be applied to students to understand the effectiveness of the method on student success.

References

1. Yaprak H, Karacı A, Demir I (2011) Prediction of the effect of varying cure conditions and w/c ratio on the compressive strength of concrete using artificial neural networks. Neural Comput Appl. doi:10.1007/s00521-011-0671-x
2. Lee SC (2003) Prediction of concrete strength using artificial neural networks. Eng Struct 25:849–857
3. Vaisla KS, Bhatt AK (2010) An analysis of the performance of artificial neural network technique for stock market forecasting. Int J Comput Sci Eng 2(6):2104–2109
4. Veronez M, de Souza S, Matsuoka M et al (2011) Regional mapping of the Geoid using GNSS (GPS) measurements and an artificial neural network. Remote Sens 3(4):668–683. doi:10.3390/rs3040668
5. Elmas Ç (2007) Yapay Zeka Uygulamaları. Seckin Yayıncılık, Ankara (in Turkish)
6. Körez A (2009) Durum Tabanlı Öğrenci Modeli İle Zeki Öğretim Sistemi(ZÖS) Tasarımı. Dissertation, University of Marmara
7. Rishi OP, Govil R, Sinha M (2007) Distributed case based reasoning for intelligent tutoring system: an agent based student modeling paradigm. World Acad Sci Eng Technol 29:273–276
8. Anohina A (2007) Advances in intelligent tutoring systems: problem-solving modes and model of hints. Int J Comput Commun Control 2(1):48–55
9. Parvez S M (2007) A pedagogical framework for integrating individual learning style into an intelligent tutoring system. Dissertation, Lehigh University
10. Butz CJ, Hua S, Maguire RB (2006) A web-based Bayesian intelligent tutoring system for computer programming. Web Intell Agent Syst Int J 4:61–81

11. Hsieh SJ, Hsieh PY (2001) Intelligent tutoring system authoring tool for manufacturing engineering education. Int J Eng Ed 17(6):569–579
12. Keleş A, Ocak R, Keleş A, Gülcü A (2009) ZOSMAT: web-based intelligent tutoring system for teaching-learning process. Expert Syst Appl 36(2):1229–1239. doi:10.1016/j.eswa.2007.11.064
13. Mitrovic A (1998) Learning SQL with a computerised tutor. In: Proceedings of 29th ACM SIGCSE Technical Symposium, Atlanta. doi:10.1145/273133.274318
14. Mayo M, Mitrovic A, McKenzie J (2000) CAPIT: an intelligent tutoring system for capitalisation and punctuation. In: Proceedings of advanced learning technology: design and development issues IEEE computer society. doi:10.1109/IWALT.2000.890594
15. Liegle OL (1999) development and evaluation of an adaptive web-based intelligent tutoring system. Dissertation, Kent State University Graduate School of Management
16. Günel K, Aşlıyan R (2009) Determining difficulty of questions in intelligent tutoring systems. Turkish Online J Educ Technol (TOJET) 8(3):1303–6521
17. Acampora G, Gaeta M, Loia V (2011) Hierarchical optimization of personalized experiences for e-Learning systems through evolutionary models. Neural Comput Appl 20:641–657. doi:10.1007/s00521-009-0273-z
18. Julie MD, Kannan B (2011) Attribute reduction and missing value imputing with ANN: prediction of learning disabilities. Neural Comput Appl. doi:10.1007/s00521-011-0619-1
19. Mullier D, Moore D, Hobbs, D (2001) A neural-network system for automatically assessing students. In: Kommers P, Richards G (eds) World conference on educational multimedia, hypermedia and telecommunications, pp 1366–1371)
20. Tortum A, Yayla N, Çelik C, Gökdağ M (2007) The investigation of model selection criteria in artificial neural networks by the Taguchi method. Phys A 386:446–468
21. Wu D (2009) Supplier selection: a hybrid model using DEA, decision tree and neural network. Expert Syst Appl 36:9105–9112. doi:10.1016/j.eswa.2008.12.039
22. Fang W, Yang B, Song D, Tang Z (2009) A new scheme on privacy-preserving distributed decision-tree mining. In: First international workshop on, education technology and computer science, 2009, ETCS '09, vol 2, pp 517–520. doi:10.1109/ETCS.2009.376