Multidisciplinary
SCIENTIFIC JOURNAL OF
MARITIME RESEARCH

University of Rijeka
Faculty of Maritime
Studies Rijeka

Multidisciplinarni
znanstveni časopis
POMORSTVO

# Cluster analysis of student activity in a web-based intelligent tutoring system

Igor Jugo[1], Božidar Kovačić[1], Edvard Tijan[2]

[1] University of Rijeka, Department of Informatics, Radmile Matejčić 2, 51000 Rijeka, Croatia
[2] University of Rijeka, Faculty of Maritime Studies Rijeka, Studentska 2, 51000 Rijeka, Croatia

ABSTRACT

In this paper we present a model of a system for integration of an intelligent tutoring system with data mining tools. The purpose of the integration is twofold; a) to power the system adaptability based on clustering and sequential pattern mining, and b) to enable teachers (non-experts in data mining) to use data mining techniques in their web browser on a daily basis, and get useful visualizations that provide insights into the learning progress of their students. We also present an approach to clustering results evaluation developed so that the system can independently deduce the best number of clusters for the k-means algorithm as well as order the clusters in terms of learning efficiency of cluster members (students).

ARTICLE INFO

Preliminary communication
Received 1 June 2015
Accepted 22 June 2015

**Key words:**
Intelligent tutoring systems
Educational data mining
Clustering students
Maritime e-learning

## 1. Introduction

Intelligent tutoring systems (ITS) have proved to be a valuable teaching tool not only for distance education but also as a complementary teaching/learning activity in traditional (face-to-face) education. Many have been developed for teaching children, especially for well-defined domains (e.g. math, geometry, etc.). Most of these tools are standalone (desktop) applications while the number of web-based ITSs if much smaller [1] especially for so called ill-defined domains [10]. Several such systems are being developed at University of Rijeka to serve as an additional learning platform on knowledge domains mainly in the field of software development (well under way) and maritime education and training (beginning phases). There are many approaches to software design and development and many approaches to software programming as well. Using this platform we can create a number of smaller knowledge domains that the students can learn as an additional part of the course or as a prerequisite for another larger learning topic. ITSs are by default adaptive learning systems as they are developed to monitor the learning progress of each student and dynamically adapt either the content structure or presentation form. In our system, as the student progresses he/she will not be presented with content they have already mastered in order to prevent boredom. ITSs also commonly give the student freedom in choosing the order in which they wish to learn the knowledge units (KUs). In that sense, the ITS does not provide any help to the student. Many authors have tried to enhance this part of an ITS by applying various machine learning and artificial intelligence methods on data recorded by the ITS. The results of these methods are often used as a basis for automatic adaptation of e-learning systems to the needs, expectations, and behaviors of their users. The primary goal of Educational Data Mining (EDM) is to use datasets from various educational systems to better understand learning and to provide information about the learning process. In one of the most cited EDM overview papers the authors put forth a set of important research objectives: “a) EDM tools have to be designed to be easier for educators or non-expert users in data mining, b) The data mining tool has to be integrated into the e-learning environment as one more traditional authoring tool, c) standardization of input data and output models, as along with preprocessing, discovering and post-processing tasks and d) Traditional mining algorithms need to be tuned to take into account the educational context” [13]. Our current and future research is concerned with the first two objectives. Performing EDM is a complex process that requires the participation of data mining (DM), database administration and other experts besides the teacher. The use of DM in e-learning systems or everyday classrooms is far from widespread. We presume that by integrating DM tools with e-learning systems (and the educational process in general) we can help significantly in broadening the application of DM in education. This integration should be seamless to the end user. In this way, the teachers could stay in the familiar environment of the Web browser and the e-learning system they regularly use, and start using new DM-powered features that will bring novel, helpful insights to their work. Another drawback is the feedback delay. Data is gathered during the semester or school year and afterwards all the steps of the DM process [19] are done by experts to finally provide the teacher with some insights. By integrating our ITS with DM tools we enable the teacher to run some DM analyses and get information about the activities and results of students he/she is currently teaching, without the need of other experts. In this paper we present a model of an integration framework to enable daily usage of clustering by teachers in the web browser and a method for conducting clustering analysis evaluation as well as to evaluate the groups of students the obtained clusters represents.

The paper is structured as follows: Section 2 introduces related work on EDM and focuses on using clustering in e-learning systems. Section 3 presents the basic functionality of our web-based ITS as well as the learning analytics module developed for teachers. The model of the integration framework is given in Section 4, while Section 5 presents the method for clustering evaluation. In Section 6 we present the results of the proposed method.

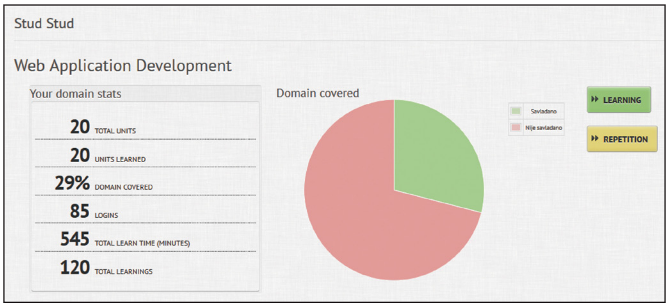
## 2. Related work

As mentioned earlier this research presents the integration of DM tools with a web-based ITS and the implementation of clustering analysis made possible through that integration. Teachers can run clustering analysis using the visual analytics module of the ITS and gain insight into the activity of students using the system. The other clustering implementation is scheduled (run at set time intervals) and is a prerequisite for sequential pattern mining implementation that will be used to help guide students through a specific knowledge domain in a more effective way (in terms of knowledge acquisition and time). Similar approaches have been applied for the purpose of recommending content web pages [14]. Romero and Ventura gave an overview of the field of EDM in 2005 [13] and in 2011 [17] in which they grouped the references by techniques and algorithms used. Romero also described the basic steps for applying common DM techniques to Moodle, a well-known course management system [15] as well as developed a Moodle block that enabled the users to perform [16] clustering, classification and association rule mining and export the output to a file. Our system integrates the results in the visual analytics module so that the teacher can continue his analysis without reading raw DM tool output. Student grouping is another important research topic in EDM. There is a large number of approaches to clustering (connectivity, centroid, distribution, density based, etc.) and an even larger number of algorithms that can be applied on student data. An overview of the clustering analysis critical steps was published by Miligan [11]. In cluster analysis, a fundamental problem is to determine the best estimate of the number of clusters, which has a deterministic effect on the clustering results. Selecting the optimal number of clusters is a well known optimization problem that has received a lot of attention. A variety of methods have been proposed to estimate the number of clusters. Gordon [4] divided these methods into two categories: global methods and local methods. With the global methods, the quality of clustering given a specific number of clusters, g, is measured by a criterion, and the optimal estimate of g, ^G, is obtained by comparing the values of the criterion calculated in a range of values of g. Some of these methods are: Calinski and Harabasz's method, Hartigan's method, Krzanowski and Lai's method, Silhouette statistic and the Gap method. Their performance has been analyzed in [21] and [20]. Finally, the obtained cluster structure can be evaluated through descriptive statistics or a number of more complex methods [4] while the interpretation depends of the research area and nature of data. In our system we rely on descriptive statistics to create an algorithm that will sort the clusters in relation to cluster members activity levels as well as learning efficiency. In this paper we also present data visualizations created using standard WWW technologies (SVG, Canvas). Other authors have developed web applications for DM based data visualizations, but rarely by using standard World Wide Web technologies. In [9] authors developed a Flex/Flash based application in the field of Bioinformatics, while in [22] authors developed a Java/Matlab based application that accepts data file uploads and returns results from a small set of DM algorithms. In [7] authors developed a student forum activity visualization tool using the Scalable Vector Graphics (SVG) web standard.

## 3. Research environment

Our web-based intelligent tutoring system (ITS) provides a platform for learning on ill-defined domains [10] i.e. domains that consist of a number of knowledge units (KUs) that do not have a strictly defined order in which they have to be taught/learned, but instead the system relies on a domain expert to define the structure of the domain. The system provides teachers with functionalities for creating KUs, teaching materials, various types of questions for assessing acquired knowledge, and an editor to create the KU hierarchy. When adding answers to questions, the teacher can define a connection between an incorrect answer and another KU if that answer is an indication of insufficient understanding of that KU. Each KU is given a start and a threshold value, which students reach by answering the questions correctly (or fall below that value by answering incorrectly).

After logging in, students are presented with a list of domains they have access to, together with basic statistical data concerning their progress through the domain, a basic visualization of the percentage of the domain the student has covered/learned, as well as action buttons for two basic actions currently at student's disposal – learning and repetition (Figure 1).



Stud Stud

Web Application Development

Your domain stats

20 TOTAL UNITS

20 UNITS LEARNED

29% DOMAIN COVERED

85 LOGINS

545 TOTAL LEARN TIME (MINUTES)

120 TOTAL LEARNINGS

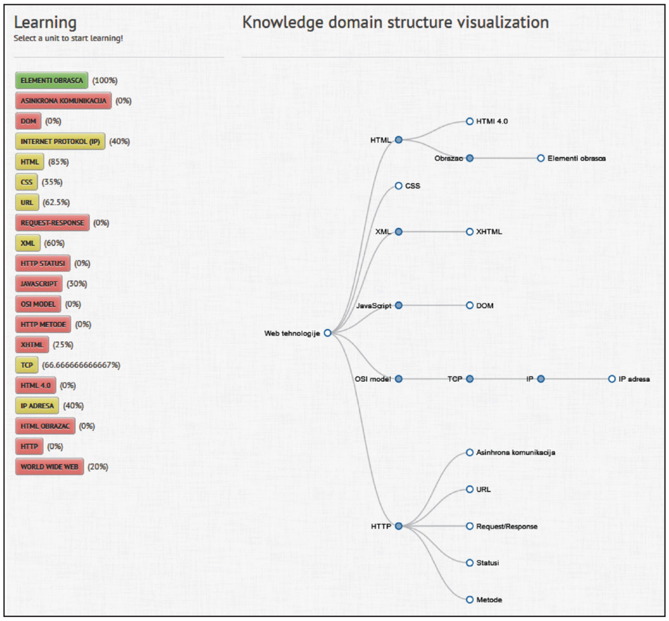
Domain covered

🟩 Savladano

🟥 Nije savladano

**Figure 1** Start page of the ITS

After selecting the "learning" action button, the student is presented with the domain structure and visual information on the knowledge units he/she has learned so far (Figure 2).



Learning

Select a unit to start learning!

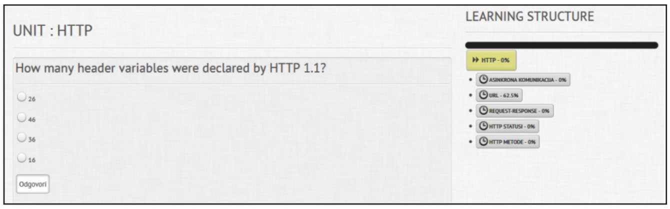
| | |
|---|---|
| ELEMENTI OBRASCA | (100%) |
| ASINKRONA KOMUNIKACIJA | (0%) |
| DOM | (0%) |
| INTERNET PROTOKOL (IP) | (40%) |
| HTML | (85%) |
| CSS | (35%) |
| URL | (62.5%) |
| REQUEST-RESPONSE | (0%) |
| XML | (60%) |
| HTTP STATUSI | (0%) |
| JAVASCRIPT | (30%) |
| OSI MODEL | (0%) |
| HTTP METODE | (0%) |
| XHTML | (25%) |
| TCP | (66.666666666667%) |
| HTML 4.0 | (0%) |
| IP ADRESA | (40%) |
| HTML OBRAZAC | (0%) |
| HTTP | (0%) |
| WORLD WIDE WEB | (20%) |

Knowledge domain structure visualization

**Figure 2** Learning starting page

By selecting one of the knowledge units, the student starts the learning process. In the next development iteration of the system, the D3JS [2] visualization on the right will incorporate all the information and hyperlinks currently on the left hand side of the screen (color, percentage of completeness and hyperlink). This should also encourage students to follow the domain structure more closely instead of advancing linearly. The first phase of the learning process is the presentation of teaching materials. Currently, the system enables the teacher to create HTML pages (with images, hyperlinks, etc.), code examples (developed system is used to teach computer science courses, maritime education and training content is still undergoing development), hyperlinks to additional learning materials (websites, PDFs, PPTs) and video lectures (currently only YouTube videos can be embedded). The system gathers data about the usage of teaching materials. This data will be used to help both students and teachers in future development iterations (e.g., popular items (for students), never used items (for the teacher)). Having read the materials, the student can move on to assess the acquired knowledge.

During assessment, the system will first ask the student a question about the KU that was displayed, followed by an initial question for every KU that is below the current KU in the domain structure. In this way the system checks whether the student understands all the underlying concepts. If the student offers an incorrect answer to any of the initial questions, he/she is transferred to learning that particular KU and the whole process is repeated. When the student has answered all the initial questions, the learning process is finished and the student can choose a different KU to learn. Once the student reaches the KU threshold, the system will stop displaying that KU later in the learning process in order to avoid tediousness and repetition. By answering questions about other KUs incorrectly, the students' knowledge level can fall below the set threshold, so the KU appears in the learning process again. Figures 3 and 4 below illustrate the learning process. The yellow button is the starting KU while the grey buttons are the KUs below it in the domain structure.



UNIT : HTTP

How many header variables were declared by HTTP 1.1?

○ 26

○ 46

○ 36

○ 16

Odgovori

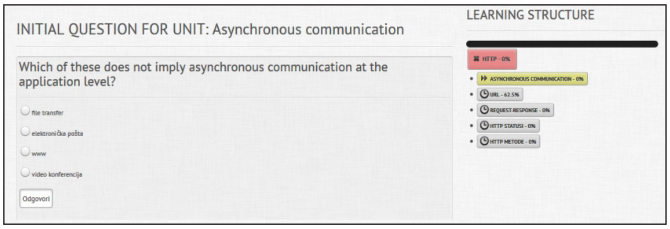
LEARNING STRUCTURE

HTTP - 0%

- ASINKRONA KOMUNIKACIJA - 0%
- URL - 62.5%
- REQUEST-RESPONSE - 0%
- HTTP STATUSI - 0%
- HTTP METODE - 0%

**Figure 3** Assessment – starting interface

Regardless of the answer to the question about the current KU, the system will ask the student one initial question about each of the KUs below.



INITIAL QUESTION FOR UNIT: Asynchronous communication

Which of these does not imply asynchronous communication at the application level?

○ file transfer

○ elektronička pošta

○ www

○ video konferencija

Odgovori

LEARNING STRUCTURE

HTTP - 0%

- ASYNCHRONOUS COMMUNICATION - 0%
- URL - 62.5%
- REQUEST-RESPONSE - 0%
- HTTP STATUSI - 0%
- HTTP METODE - 0%

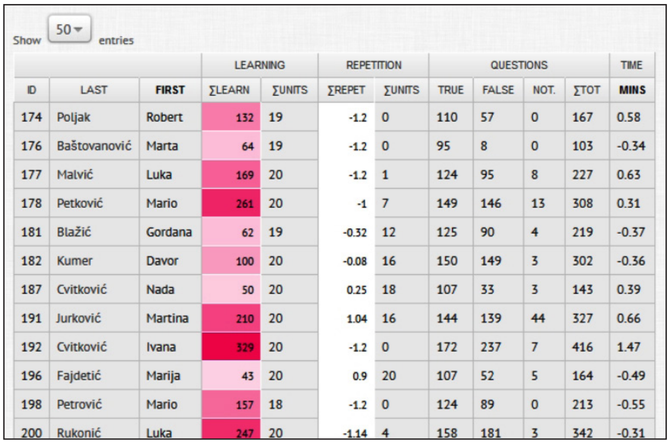
**Figure 4** Assesment – testing knowledge on sub. KUs

No matter how many levels down the hierarchy the student is taken by answering initial questions incorrectly, the system will always return to the starting level and finish when all the initial questions have been answered. After the learning process is finished, the student is presented with a visualization of his/hers results on all the units learned during the process.

The other main functionality for the student is the repetition process which provides the student with a mechanism for answering a number of questions about the selected KU without presenting learning materials. In this way the student can reach the unit threshold faster. Students can also try "gaming" the system by answering a lot of questions in a very short period of time in order to extract as many questions from the database (question selection process follows a number of preconditions in order to present another question each time) or to guess the correct answer. This kind of behavior was expected and confirmed by analyzing usage data. In the next development iteration we will make a number of changes that will discourage this behavior, enable the system to recognize and react accordingly, as well as develop early warning indicators for teachers.

Data visualization is closely connected with statistical analysis and data mining. Through data visualization we tap into the vast human potential for spotting patterns, identifying exceptions and important variations in data that would be overlooked in tabular form. The output of DM methods is usually displayed in two-dimensional charts, but with new web technologies we can also create trees, network maps, animations (time lapse, heat maps, path/pattern following, etc.) and add interactivity (zoom in/out for general->detailed views, etc.).

We developed a fully customizable, browser-based, visual analytics module for teachers in order to give them useful insights into the activity of the students and the learning process as a whole. The analytics is powered by continuous aggregation and statistical analysis of the data, as well as by integration with data mining tools and web-standards-compliant data visualization frameworks. When they start the analytics module, teachers are presented with a compact report (Figure 5) with heat maps in some columns.

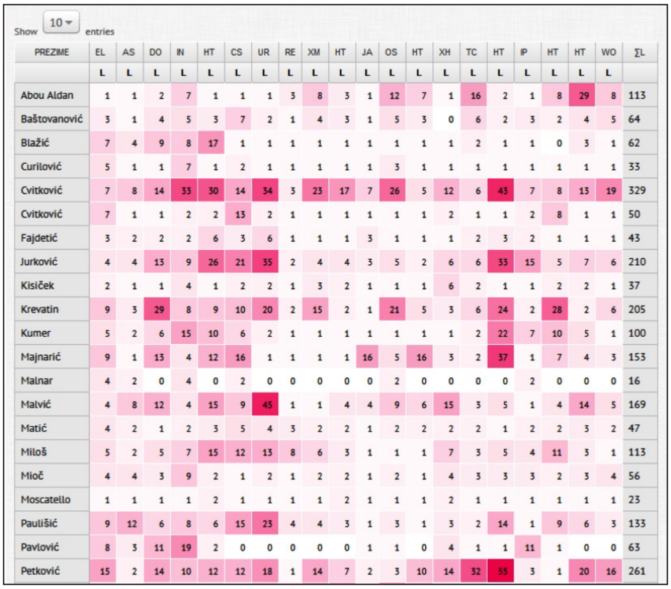


Show 50 entries

| | | | LEARNING | | REPETITION | | QUESTIONS | | | | TIME |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | LAST | FIRST | ΣLEARN | ΣUNITS | ΣREPET | ΣUNITS | TRUE | FALSE | NOT. | ΣTOT | MINS |
| 174 | Poljak | Robert | 132 | 19 | -1.2 | 0 | 110 | 57 | 0 | 167 | 0.58 |
| 176 | Baštovanović | Marta | 64 | 19 | -1.2 | 0 | 95 | 8 | 0 | 103 | -0.34 |
| 177 | Malvić | Luka | 169 | 20 | -1.2 | 1 | 124 | 95 | 8 | 227 | 0.63 |
| 178 | Petković | Mario | 261 | 20 | -1 | 7 | 149 | 146 | 13 | 308 | 0.31 |
| 181 | Blažić | Gordana | 62 | 19 | -0.32 | 12 | 125 | 90 | 4 | 219 | -0.37 |
| 182 | Kumer | Davor | 100 | 20 | -0.08 | 16 | 150 | 149 | 3 | 302 | -0.36 |
| 187 | Cvitković | Nada | 50 | 20 | 0.25 | 18 | 107 | 33 | 3 | 143 | 0.39 |
| 191 | Jurković | Martina | 210 | 20 | 1.04 | 16 | 144 | 139 | 44 | 327 | 0.66 |
| 192 | Cvitković | Ivana | 329 | 20 | -1.2 | 0 | 172 | 237 | 7 | 416 | 1.47 |
| 196 | Fajdetić | Marija | 43 | 20 | 0.9 | 20 | 107 | 52 | 5 | 164 | -0.49 |
| 198 | Petrović | Mario | 157 | 18 | -1.2 | 0 | 124 | 89 | 0 | 213 | -0.55 |
| 200 | Rukonić | Luka | 247 | 20 | -1.14 | 4 | 158 | 181 | 3 | 342 | -0.31 |

**Figure 5** Compact report for selected group

The report contains aggregate data on the number of learning sessions, KUs learned, repetitions, KUs repeated, questions answered correctly, incorrectly, and not answered, total number of presented questions and total time spent learning (in minutes). Each row presents data about a single student. Each of the columns can be expanded to get a detailed view about the student's activity. Figure 6 represents the expanded report on the number of learning sessions and repetitions for all the KUs in the domain.



Show 10 entries

| PREZIME | EL | AS | DO | IN | HT | CS | UR | RE | XM | HT | JA | OS | HT | XH | TC | HT | IP | HT | HT | WO | ΣL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | L | L | L | L | L | L | L | L | L | L | L | L | L | L | L | L | L | L | L | L | |
| Abou Aldan | 1 | 1 | 2 | 7 | 1 | 1 | 1 | 3 | 8 | 3 | 1 | 12 | 7 | 1 | 16 | 2 | 1 | 8 | 29 | 8 | 113 |
| Baštovanović | 3 | 1 | 4 | 5 | 3 | 7 | 2 | 1 | 4 | 3 | 1 | 5 | 3 | 0 | 6 | 2 | 3 | 2 | 4 | 5 | 64 |
| Blažić | 7 | 4 | 9 | 8 | 17 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 0 | 3 | 1 | 62 |
| Curilović | 5 | 1 | 1 | 7 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 33 |
| Cvitković | 7 | 8 | 14 | 33 | 30 | 14 | 34 | 3 | 23 | 17 | 7 | 26 | 5 | 12 | 6 | 43 | 7 | 8 | 13 | 19 | 329 |
| Cvitković | 7 | 1 | 1 | 2 | 2 | 13 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 8 | 1 | 1 | 50 |
| Fajdetić | 3 | 2 | 2 | 2 | 6 | 3 | 6 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 3 | 2 | 1 | 1 | 1 | 43 |
| Jurković | 4 | 4 | 13 | 9 | 26 | 21 | 35 | 2 | 4 | 4 | 3 | 5 | 2 | 6 | 6 | 33 | 15 | 5 | 7 | 6 | 210 |
| Kisiček | 2 | 1 | 1 | 4 | 1 | 2 | 2 | 1 | 3 | 2 | 1 | 1 | 1 | 6 | 2 | 1 | 1 | 2 | 2 | 1 | 37 |
| Krevatin | 9 | 3 | 29 | 8 | 9 | 10 | 20 | 2 | 15 | 2 | 1 | 21 | 5 | 3 | 6 | 24 | 2 | 28 | 2 | 6 | 205 |
| Kumer | 5 | 2 | 6 | 15 | 10 | 6 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 22 | 7 | 10 | 5 | 1 | 100 |
| Majnarić | 9 | 1 | 13 | 4 | 12 | 16 | 1 | 1 | 1 | 1 | 16 | 5 | 16 | 3 | 2 | 37 | 1 | 7 | 4 | 3 | 153 |
| Malnar | 4 | 2 | 0 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 16 |
| Malvić | 4 | 8 | 12 | 4 | 15 | 9 | 45 | 1 | 1 | 4 | 4 | 9 | 6 | 15 | 3 | 5 | 1 | 4 | 14 | 5 | 169 |
| Matić | 4 | 2 | 1 | 2 | 3 | 5 | 4 | 3 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 3 | 2 | 47 |
| Miloš | 5 | 2 | 5 | 7 | 15 | 12 | 13 | 8 | 6 | 3 | 1 | 1 | 1 | 7 | 3 | 5 | 4 | 11 | 3 | 1 | 113 |
| Mioč | 4 | 4 | 3 | 9 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 4 | 3 | 3 | 3 | 2 | 3 | 4 | 56 |
| Moscatello | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 23 |
| Paulišić | 9 | 12 | 6 | 8 | 6 | 15 | 23 | 4 | 4 | 3 | 1 | 3 | 1 | 3 | 2 | 14 | 1 | 9 | 6 | 3 | 133 |
| Pavlović | 8 | 3 | 11 | 19 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 4 | 1 | 1 | 11 | 1 | 0 | 0 | 63 |
| Petković | 15 | 2 | 14 | 10 | 12 | 12 | 18 | 1 | 14 | 7 | 2 | 3 | 10 | 14 | 32 | 55 | 3 | 1 | 20 | 16 | 261 |

**Figure 6** Expended report on learnings for each KU

The same can be done with questions and learning time columns. The columns represent each KU (full names are revealed in tooltips), while rows represent students. The table is interactive – the teacher can define the number of rows to be displayed, search the table, sort by any column, etc. This heat map can reveal which KUs the students found particularly easy or difficult, or which students had the most difficulties to progress through the domain. Another part of the visual analytics module is the chart section. There is a number of activity charts (e.g., day-by-day activity or cumulative day-by-day activity) that can reveal the activity levels of the whole group or individual students (Figure 7).

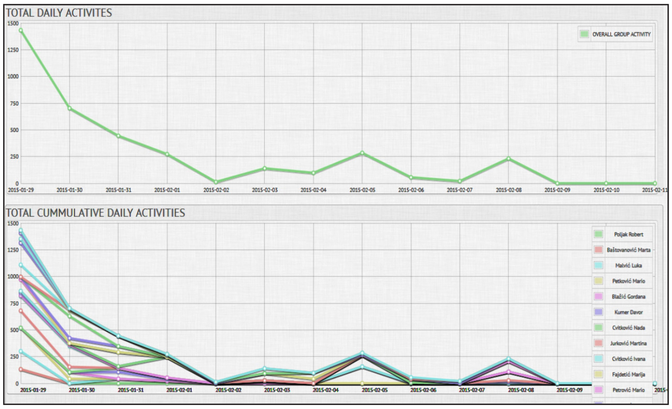


**Figure 7** Total and cumulative daily activity chart

From the compact, expanded or chart reports, the teacher can select any student and review his/hers actions, activity and learning progress.

## 4. Model for ITS and DM tools integration

In order to reach the objectives mentioned in Section 1 we propose a solution that integrates our web-based ITS with standalone data mining tools – Weka [5] and SPMF [3]. Other researchers have described some form of integration of data mining algorithms and web applications, but most of them are either Java web applications or they hardcode the implementation of a single algorithm into their system. Another implementation problem is the communication between our ITS (which is a PHP web application) and DM tools written in Java. We wanted to develop an integration module that will enable continuous communication with the DM tools without re-implementing any specific algorithm into our application or changing the original DM tool. In this way the data from our system can be analyzed by a DM expert on another machine, running the same DM tools, with absolute confidence that the results will be the same (where it is possible, depending on the algorithm). Figure 8 presents the architecture of the integrated system. Functionalities that rely on data mining results for students and teachers are marked with asterisks. As mentioned earlier, the integration enables us to do DM analyses either automatically (using scheduled cron jobs) or on-demand by the teacher. The first DM-powered analysis for the teacher analytics module is clustering. We will use this analysis to describe the system architecture and the process of acquiring clustering information from the DM tools.

We have described the student and teacher interfaces in the previous Section. Clustering for students will be done automatically together with sequential pattern mining algorithm that will enable us to dynamically modify the learning structure (presented in Figures 3 and 4) based on his/hers current knowledge level, activity level, learning paths and efficiency. Clustering for teachers will be done on-demand in order to help them differentiate between more and less active/successful groups of students.

The scheduled clustering analysis is based on a fixed number of features (see Table 1), while on-demand analysis enables the teacher to choose features.

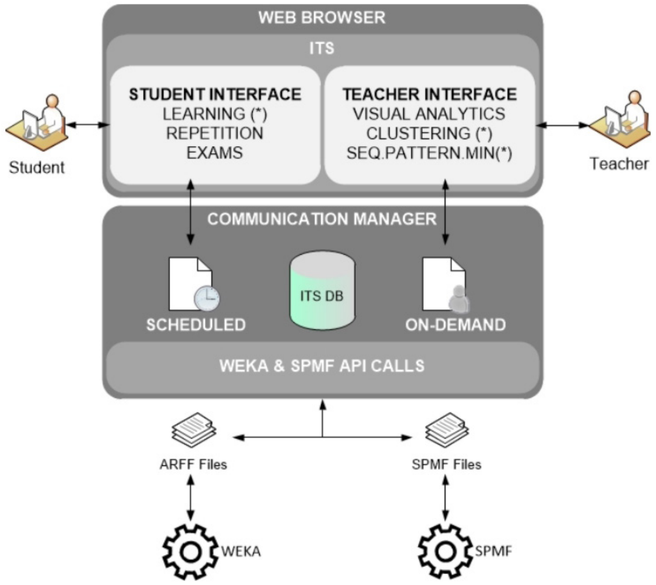


**Figure 8** Overall system architecture

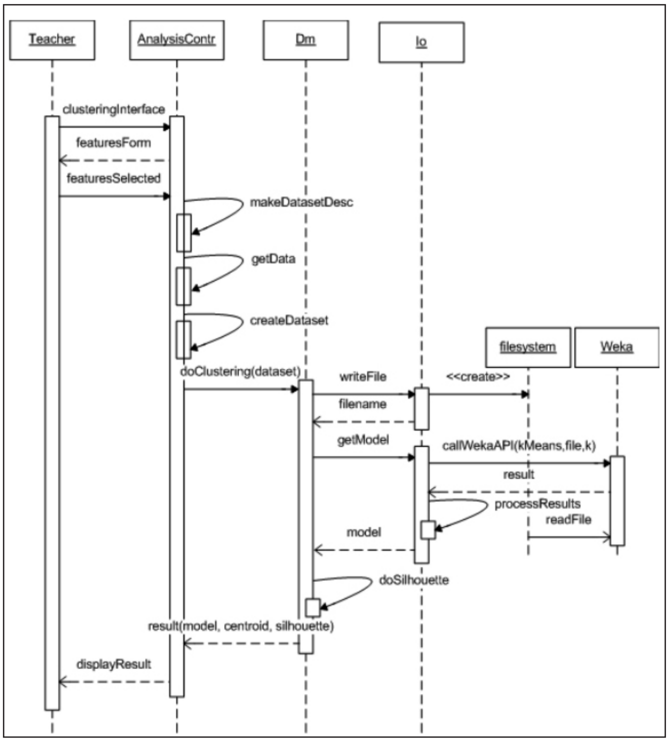


**Figure 9** UML sequence model for one clustering analysis

The UML sequence model for the clustering analysis is displayed in Figure 9. The communication manager servers as a bridge between the ITS and the DM tools. It gathers data and the stored system settings (tool-algorithm-file format mappings) from the ITS database. It is responsible for data retrieval, processing, formatting and creating DM tool API calls as well as working with the servers file system. When the teacher selects the clustering tab, the system invokes the communication manager by sending the clustering keyword, data and data description. The communication manager converts the data to the appropriate file format (ARFF or SPMF depending on the keyword-to-tool mapping), writes the file to the file system and then performs the appropriate API call in the shell command line.

Based on the current functionalities of the system we created a set of engineered features that was used for clustering analysis. This set was developed using the learning data obtained from the database in order to better represent the **current** activity levels and efficiency of students. The list of engineered features will be expanded in the future to create more precise models.

**Table 1** Feature sets for clustering analysis

| **Engineered features $F_{EF}$** | |
|---|---|
| Total number of **learning** actions * percentage of knowledge domain covered (standardized) | $L\%_{std}$ |
| Total number of **repetition** actions * percentage of knowledge domain covered (standardized) | $R\%_{std}$ |
| Total **time** spent learning * percentage of knowledge domain covered (standardized) | $T\%_{std}$ |
| **Effectiveness** on completed KUs | $E1_{KUi}$ |
| Effectiveness on uncompleted KUs where minimum number of questions needed to cover the KU was not surpassed | $E2_{KUi}$ |
| Effectiveness on uncompleted KUs where minimum number of questions needed to cover the KU was surpassed | $E3_{KUi}$ |
| Overall student effectiveness E1+E2+E3 standardized (x-μ)/σ | $E_{totalstd}$ |

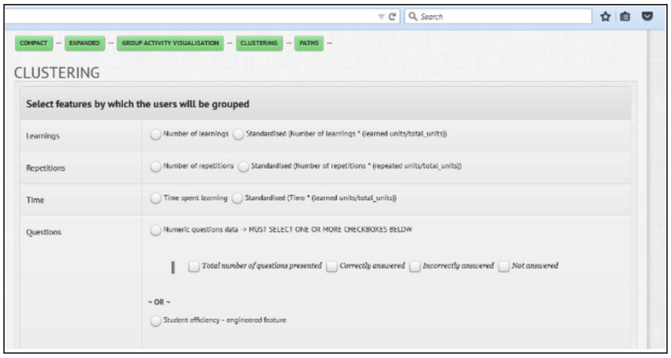
The detailed algorithm of the on-demand functionality is presented below. The algorithm set the initial value k for kMeans using the simple "elbow" method. The final number of loop iterations is not important as the loop will break as soon as we get a model that contains any cluster size of 1, as we are interested in larger clusters. For each clustering we also perform the clustering evaluation using the silhouette statistic (details in next Section) to provide the teacher with additional information about the quality of distribution. The algorithm is shown below:

```
Algorithm 1: on-demand clustering analysis

Input: domain identifier Di, domain group identifier $DG_j$
       Set of features F selected by user
       Clustering system settings:
              tool T ⍰ [weka,spmf]
              format FT ⍰ [arff,spmf]
Output: k [centroids, members, silhouette]

retrieve learning data
for each $f \in (F_{EF})$ calculate and standardize value
create dataset ($D_i$, $DG_j$, F)
write dataset to file ($D_i$, $DG_j$, time, FT) => filename
set initial k = sqrt(sizeof(dataset)/2)
results[]
for i=2 to k+5
    [centroids, model] =
        construct clustering api call (T,i, dataset)
        perform api call (i, dataset)
        process api call output
    silhouette(i, model)
    clusterOrder(model, centroids) // see Alg. 2
    results[i]=centroids, model, silhouette
    if (number of clusters in model with size == 1)>=2)
        break
return results
```

In the scheduled scenario the clustering analysis uses the engineered feature set $F_{EF}$. The feature set is chosen and can be edited by the system administrator. Another change is that the scheduled analysis returns only one model – the model with the highest silhouette statistic. The user interface for feature selection and starting the clustering analysis is displayed in Figure 10.


Search
COMPACT -- EXPANDED -- GROUP ACTIVITY VISUALISATION -- CLUSTERING -- PATHS --

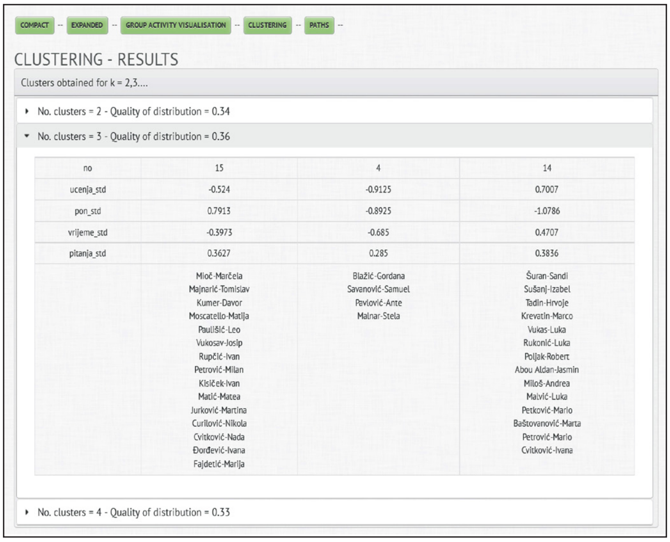
CLUSTERING

**Select features by which the users will be grouped**

| | |
|---|---|
| Learnings | ○ Number of learnings ○ Standardised (Number of learnings * (learned units/total_units)) |
| Repetitions | ○ Number of repetitions ○ Standardised (Number of repetitions * (repeated units/total_units)) |
| Time | ○ Time spent learning ○ Standardised (Time * (learned units/total_units)) |
| Questions | ○ Numeric questions data -> MUST SELECT ONE OR MORE CHECKBOXES BELOW<br>□ *Total number of questions presented* □ *Correctly answered* □ *Incorrectly answered* □ *Not answered*<br>~ OR ~<br>○ Student efficiency - engineered feature |

**Figure 10** On demand clustering – feature selection

The interface displaying the different obtained models can be seen in Figure 11.


COMPACT -- EXPANDED -- GROUP ACTIVITY VISUALISATION -- CLUSTERING -- PATHS --

CLUSTERING - RESULTS

Clusters obtained for k = 2,3....

▸ No. clusters = 2 - Quality of distribution = 0.34

▾ No. clusters = 3 - Quality of distribution = 0.36

| no | 15 | 4 | 14 |
|---|---|---|---|
| ucenja_std | -0.524 | -0.9125 | 0.7007 |
| pon_std | 0.7913 | -0.8925 | -1.0786 |
| vrijeme_std | -0.3973 | -0.685 | 0.4707 |
| pitanja_std | 0.3627 | 0.285 | 0.3836 |
| | Mioč-Marčela<br>Majnarić-Tomislav<br>Kumer-Davor<br>Moscatello-Matija<br>Paulišić-Leo<br>Vukosav-Josip<br>Rupčić-Ivan<br>Petrović-Milan<br>Kisiček-Ivan<br>Matić-Matea<br>Jurković-Martina<br>Curilović-Nikola<br>Cvitković-Nada<br>Đorđević-Ivana<br>Fajdetić-Marija | Blažić-Gordana<br>Savanović-Samuel<br>Pavlović-Ante<br>Malnar-Stela | Šuran-Sandi<br>Sušanj-Izabel<br>Tadin-Hrvoje<br>Krevatin-Marco<br>Vukas-Luka<br>Rukonić-Luka<br>Poljak-Robert<br>Abou Aldan-Jasmin<br>Miloš-Andrea<br>Malvić-Luka<br>Petković-Mario<br>Baštovanović-Marta<br>Petrović-Mario<br>Cvitković-Ivana |

▸ No. clusters = 4 - Quality of distribution = 0.33

**Figure 11** On demand clustering – results

From this point the teacher can select any of the students to analyze his/hers learning activities using our visual analytics, or display detailed information (feature values) for each student. In the next section we present our approaches to evaluating the cluster distribution using the silhouette statistic and evaluating the cluster learning activity.

## 5. Proposed model for clustering results evaluation

In order to provide a quality metric for both on-demand and scheduled analysis scenario we needed to implement an evaluation method. As mentioned in Section 2, many methods have been developed and tested by researchers [20, 21]. Some of these methods require changes to the clustering algorithm source while others can be applied after the model was acquired. We chose to implement the silhouette statistic [8] due to the fact that it does not require us to change the implementation of the k-means algorithm as we set a goal to use the DM tools and not modify them in order to make sure that analysis on another computer with the same DM tools will produce the same results.

The definition of the silhouette statistic is based on the silhouettes introduced by Rousseeuw [18], which are constructed to show graphically how well each object is classified in a given clustering output. To plot the silhouette of the *m*th cluster, for each object in C*m*, calculate s(i) as

$a(i)$ = average dissimilarity of object i to all other objects in the *m*th cluster
$d(i;C)$ = average dissimilarity of object i to all other objects in cluster C; $C \neq Cm$
$b(i) = \min d(i;C)$
$s(i) = b(i) - a(i)/\max\{a(i); b(i)\}$

The silhouette statistic, denoted by s(g), is defined as the average of the s(i) for all objects in the data. s(g) is called the average silhouette width for the entire data set, reflecting the within-cluster compactness and between-cluster separation of a clustering. Compute s(g) for g = 1, 2, …. The optimum value of g is chosen such that s(g) is maximized over all g: ^G = arg maxgs(g).

From the above definition it is clear that the possible values for s(g) are: $-1 < s(g) < 1$. Average s(g) over all data of a cluster is a measure of how tightly grouped all the data in the cluster are. Thus the average s(g) over all data of the entire dataset is a measure of how appropriately the data has been clustered. The silhouette method is called in line 12 of Algorithm 1.

The second part of the evaluation process starts when the best clustering model (the model with the highest s(g)) is found. We need to order the clusters by the overall learning efficiency of cluster members. This part of the evaluation is more significant for the scheduled clustering analysis although it is a useful indicator for the teacher.

The ordering of clusters will be used in the next phase of our research as a component of learning path efficiency evaluation. In this way we will be able to link students learning paths with the efficiency of the cluster the student belongs to and suggest these learning paths to members of a less efficient cluster.

After analyzing the learning data gathered during system testing, we created visualizations of value distributions of each of the engineered features datasets. All of the distributions we are very close to a normal distribution. That enabled us to use standard deviation intervals to define a set of scores. We then developed score-to-interval mappings for each feature as displayed in Table 2.

**Table 2** Score-to-interval mappings for $F_{EF}$

| SD | -3 | -2 | -1 | -0,5 | 0,5 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|
| **$L\%_{std}$** | 8 | 6 | 1 | 2 | 3 | 4 | 5 | 7 |
| **$R\%_{std}$** | 8 | 6 | 1 | 2 | 3 | 4 | 5 | 7 |
| **$T\%_{std}$** | 8 | 2 | 1 | 3 | 4 | 5 | 6 | 7 |
| **$E_{total}$** | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

An algorithm was developed to dynamically determine the interval values and calculate the overall cluster learning efficiency ordering.

```
Algorithm 2: clusterOrder (scheduled analysis)
Input:  model (student data with cluster assignments)
        cluster centroids
Output: cluster efficiency ordering

clusterOrder[], centroidScores[]
for each F ∈ ($F_{EF}$)
  get score-to-interval mapping M(f)
  calculate $\mu$ (model(f))
  calculate $\sigma$ (model(f))
  calculate interval values
  for each  m ∈ M(f)
    for each f ∈ F // centroid value
      check interval
        get score
        centroidScores[f]=score
for each cs ∈ centroidScores
  clusteringOrder[] = sum(cs)
sort(clusteringOrder[])
return clusteringOrder
```

The results of the clustering evaluation is described in the next Section.

## 6. Results and future work

To test the clustering functionality and the values of the silhouette statistic we used the data collected from a knowledge domain we developed for the third year undergraduate students. The domain consisted of twenty knowledge units ($KU_{TOTAL}=20$). The students had access to the domain for 10 days. The domain was used by three different groups of students. The basic statistics are presented in Table 2.

**Table 3** Basic statistic on student groups

| Groups | G1 | G2 | G3 |
|---|---|---|---|
| Students | 33 | 31 | 11 |
| Active | 33 | 30 | 10 |
| Average % completed | 97% | 98% | 90% |

To test the algorithm for optimal k value selection based on the silhouette statistic we ran the clustering analysis for all groups using the engineered features $F_{EF}$ set. Due to the size limitations of this paper we present the results for group 1 only in Table 4 below.

**Table 4** Silhouette statistic and cluster sizes for G1

| K | S(k) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0,34 | 16 | 17 | | | | | | | |
| 3 | 0,36 | 15 | 4 | 14 | | | | | | |
| 4 | 0,33 | 11 | 2 | 8 | 11 | | | | | |
| 5 | **0,47** | 12 | 2 | 7 | 11 | 1 | | | | |
| 6 | 0,45 | 9 | 2 | 7 | 9 | 1 | 5 | | | |
| 7 | 0,38 | 6 | 2 | 7 | 8 | 1 | 5 | 4 | | |
| 8 | 0,29 | 6 | 2 | 7 | 7 | 1 | 3 | 4 | 3 | |
| 9 | 0,42 | 6 | 2 | 6 | 7 | 1 | 3 | 4 | 3 | 1 |

We analyzed the student data behind clustering distributions that had the highest silhouette scores and verified the results. The cluster ordering algorithm was used to calculate the learning efficiency ordering of clusters for the solution with the highest silhouette statistic value (k=5). The results for group 1 are presented in Table 5.

**Table 5** Cluster ordering results for group 1

| **G1** | **k** | | | | | **SD** |
|---|---|---|---|---|---|---|
| **Centroid** | **1** | **2** | **3** | **4** | **5** | |
| $L\%_{std}$ | -0,598 | -1,150 | 1,494 | -0,220 | -0,260 | 1,000 |
| $R\%_{std}$ | 0,980 | -1,025 | -1,100 | -0,692 | -1,200 | 0,990 |
| $T\%_{std}$ | -0,488 | -0,845 | 0,329 | -0,048 | 3,660 | 0,940 |
| $E_{totalstd}$ | 0,352 | 0,155 | 0,280 | 0,446 | 0,560 | 0,130 |
| $L_{score}$ | 1 | 6 | 6 | 2 | 2 | |
| $R_{score}$ | 5 | 1 | 1 | 2 | 6 | |
| $T_{score}$ | 3 | 1 | 4 | 3 | 7 | |
| $E_{score}$ | 4 | 6 | 5 | 4 | 3 | |
| Total | 13 | 14 | 16 | 11 | 18 | |
| Ordering | 2 | 3 | 4 | 1 | 5 | |

The results showed that the cluster ordering algorithm sorted the clusters correctly. As the scheduled clustering analysis is run at set intervals (e.g. every 3,6,12 hours) from the day the students started using the system we did encounter situations where the distributions of feature values in the model was not close to normal. As displayed in Figure 7, students created almost 40% of records in the total dataset on the first day so those situations will not happen often.

In our future work we will implement normal distribution tests like Jarque-Bera, Shapiro-Wilk or Anderson-Darling [6] into our algorithm along with a solution for scoring not-normal distributions. Another goal for future work is finer scale scoring and development of new engineered features that will represent students learning efficiency even more precisely. We will also evaluate our algorithms on a larger number of knowledge domains and a larger student groups in different fields as well as test other methods of cluster evaluation mentioned in the paper.

## 7. Conclusion

Student clustering is a common task in educational data mining and commonly a first phase of the recommender module in e-learning applications that rely on data mining.

We presented and implemented an architecture that enables our intelligent tutoring systems to communicate with data mining tools without help from other experts and without the time delay that is often present in applications that use educational data mining. Our system enables teachers to analyze the learning data using various visual analytics as well as run clustering analysis from their web browser. Besides on-demand clustering our integration architecture enables the system to run scheduled clustering analysis that is a prerequisite to the next part of our research that will use sequential pattern mining algorithms to find and evaluate common learning paths and use them to increase the system adaptability. In order to guide students towards more efficient paths through the knowledge domain we need to know to which cluster of students does the student belong and how well does that cluster perform in relation to others. The second part of this work presents a model for selecting an optimal number of clusters and determining the learning efficiency of each of the clusters in the selected clustering model. Although these issues could be easily solved by an expert, we need our system to function independently and update the database with new results at scheduled intervals.

## Acknowledgment

This research is a part of the Project "Enhancing the efficiency of an e-learning system based on data mining", code: 13.13.1.2.02., funded by the University of Rijeka, Croatia.

## References

[1] Brusilovsky, P. "Adaptive and intelligent technologies for web-based eduction." KI 13.4 (1999):19–25.

[2] D3js.org. The D3 JavaScript Library, http://d3js.org/, accessed: May 10th 2015.

[3] Fournier-Viger, P. et al., "SPMF: a Java Open-Source Pattern Mining Library" in J. Machine Learning Research, 2014, vol. 15, pp. 3389–3393.

[4] Gordon, A. D., Classification. Chapman & Hall/CRC, Boca Raton, FL, 2nd edition, 1999.

[5] Hall, M. et al., "The WEKA Data Mining Software: An Update" in SIGKDD Explorations, 2009, vol. 11, issue 1.

[6] Jarque, C. M.; Bera, A. K., "Efficient tests for normality, homoscedasticity and serial independence of regression residuals". Economics Letters, 1980;6(3):255–259.

[7] Jyothi, S, McAvinia, C., Keating, J; A visualisation tool to aid exploration of students' interactions in asynchronous online communication. Computers & Education, 58(1):30–42, 2012.

[8] Kaufman, L., Rousseeuw, P. J., Finding Groups in Data. An Introduction to Cluster Analysis. Wiley-Interscience, New York, 1990.

[9] Koelling, J; Langenkaemper, D; Abouna, S; et al.; WHIDE-a web tool for visual data mining colocation patterns in multivariate bioimages. Bioinformatics, 28(8):1143–1150, 2012.

[10] Lynch, C. et al., "Defining Ill-Defined Domains; A literature survey" in Proc. Intelligent Tutoring Systems Ill-Defined Domains Workshop, Taiwan, 2006, pp. 1–10.

[11] Milligan, G. W., A validation study of a variable weighting algorithm for cluster analysis. Journal of Classification, 6:53–71, 1989.

[12] Oztuna D, Elhan AH, Tuccar E. Investigation of four different normality tests in terms of type 1 error rate and power under different distributions. Turkish Journal of Medical Sciences. 2006;36(3):171–6.

[13] Romero, C., Ventura, S., Educational Data Mining: a Survey from 1995 to 2005, Expert Systems with Applications, vol. 1, no. 33, pp. 135–146, 2007.

[14] Romero, C., et al. "Personalized links recommendation based on data mining in adaptive educational hypermedia systems". Creating New Learning Experiences on a Global Scale. Springer Berlin Heidelberg, 2007, pp. 292–306.

[15] Romero, C., Ventura, S., García, E. "Data mining in course management systems: Moodle case study and tutorial". Computers & Education 51.1 (2008): 368–384.

[16] Romero, C., Castro, C., Ventura., S., "A Moodle Block for Selecting, Visualizing and Mining Students' Usage Data". Educational Data Mining 2013.

[17] Romero, C., & Ventura, S., Educational data mining: a review of the state of the art. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40(6):601–618, 2010.

[18] Rousseeuw, P. J., Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20:53–65, 1987.

[19] Shearer C., The CRISP-DM Model: The New Blueprint for Data Mining, Journal of Data Warehousing, vol. 5, no. 4, pp. 13–22, 2000.

[20] Symons. M. J., Clustering criteria and multivariate normal mixtures. Biometrics, 37:35–43, 1981.

[21] Tibshirani, R., Walther, G., Hastie, T., Estimating the number of data clusters via the gap statistic. Journal of the Royal Statistical Society B, 63:411–423, 2001.

[22] Zorrilla, M, García-Saiz, D; A service oriented architecture to provide data mining services for non-expert data miners. Decision Support Systems, 55(1):399–411, 2013.