Original software publication

UBUMonitor: Desktop application for visual e-learning student clustering with Moodle

Raúl Marticorena-Sánchez, Carlos López-Nozal, Ana Serrano-Mamolar *, Alicia Olivares-Gil

Universidad de Burgos, Department of Computer Science, Avda. Cantabria s/n, 09006 Burgos, Spain

ARTICLE INFO

ABSTRACT

UBUMonitor is an open-source Java desktop tool designed to analyse student interactions and grades within courses in the Moodle learning management system. Its primary goal is to assist educational researchers in making informed decisions. The application offers flexible configuration of educational datasets, enabling educational researchers to customize features of their analyses based on filters such as time, participants, e-resources, e-forums and teaching e-activities. With a modular architecture developed iteratively in Github with 35 releases, UBUMonitor has demonstrated its usability in teaching through several published use cases. This paper focuses specifically on the student clustering module, which allows customization of course activity access and grading, supports multiple clustering algorithms, and incorporates techniques for validating both the optimal number of clusters and the quality of cluster result.

Code metadata

Current code version	2.10.3
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00802
Permanent link to Reproducible Capsule	UBUMonitorLauncher with the exclusive code for checking new versions, downloading them, and launching the application
	https://github.com/yjx0003/UBUMonitorLauncher/releases/tag/v1.0.1
Legal Code License	MIT License.
Code versioning system used	git
Software code languages, tools, and services used	Java, JavaScript, JavaFX, CSS, HTML
Compilation requirements, operating environments & dependencies	Maven 4.0.0. Windows, GNU/Linux o MacOS.
If available Link to developer documentation/manual	https://ubumonitordocs.readthedocs.io/en/latest/clustering/index.html
Support email for questions	rmartico@ubu.es

Software metadata

Current software version	v2.10.3
Permanent link to executables of this version	https://github.com/yjx0003/UBUMonitor/releases/
Permanent link to Reproducible Capsule	UBUMonitorLauncher with the exclusive code for checking new versions, downloading them, and launching the application
	https://github.com/yjx0003/UBUMonitorLauncher/releases/tag/v1.0.1
Legal Software License	MIT License.
Computing platforms/Operating Systems	Windows, GNU/Linux o Mac.
Installation requirements & dependencies	Multiplatform installer UBUMonitorLauncher without dependencies.
If available, link to user manual - if formally published include a reference to the publication in the reference list	https://ubumonitordocs.readthedocs.io/en/latest/install.html
Support email for questions	rmartico@ubu.es

---

* Corresponding author.
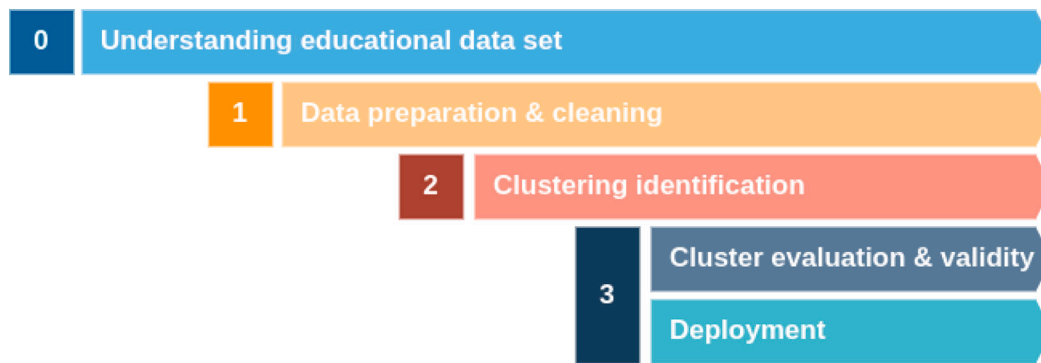 *E-mail address:* asmamolar@ubu.es (Ana Serrano-Mamolar).

Fig. 1. Educational Data Clustering process presented in *A systematic review on educational data mining* [3].

1. Motivation and significance

In the field of educational research, Learning Analytics (LA) has a significant impact on the improvement of teaching and learning processes. From its conception LA has been concerned with solving the problems associated with the growth in the availability, quantity, speed and type of data in learning environments [1]. The predictive modelling of educational data mining (EDM) and learning analytics in teaching and learning has become a core practice of researchers, largely focused on predicting student success, operationalized by academic performance [2]. Both fields, LA an EDM, require progress in the development of specialized software applications that retrieve available data on student activity in the virtual courses of a Learning Management System (LMS) and implement analysis techniques.

UBUMonitor is an educator-oriented desktop software for monitoring student learning activity in Moodle courses. In particular, this software unleashes its full potential in the hands of an educational researcher. In this paper, this role refers to educators who design learning activities within Moodle courses, evaluating them with real students and conducting empirical analyses of the level of student interaction and its direct impact on the learning process. Moreover, these educators hold elementary knowledge of the Educational Data Clustering (EDC) process [3]. UBUMonitor uses all the information collected by Moodle in order to enable educational researchers to setup their analysis of student learning patterns. This tool is instrumental in identifying the diverse groups of students, enabling educational researchers to gain comprehensive insights into their learning dynamics and thereby adapt teaching methodologies to meet their varied needs and preferences. Clustering is one of the most important techniques in the field of Data Mining and UBUMonitor implements multiple clustering algorithms so that research educators can use them to extract knowledge about the learning process of students.

Clustering is a machine learning technique used to group similar instances into distinct subsets, called clusters, based on the similarity of their features. It is unsupervised, meaning that the model has no prior knowledge of the cluster to which each instance belongs [4]. In our educational context, the instances correspond to the students and the features to their interactions in the Moodle LMS. UBUMonitor enables educational researchers to choose between the following clustering algorithms: *k*-means, *k*-means++, *x*-means, *g*-means, fuzzy clustering, DBSCAN, and DENCLUE. UBUMonitor's clustering module provides metrics and helps validate the silhouette coefficient, elbow method and dendrogram visualization of hierarchical clustering. The silhouette coefficient is an internal validation metric that measures how closely the instances are clustered. It is applied to each instance and consists of computing the distance between the clusters and the instance. The elbow method is an analysis to determine the optimal number of clusters. It is based on computing the variance within each cluster. To obtain a graphical representation of the cluster in 2D and 3D, the PCA algorithm is used to reduce the dimensionality of the features.

A dendrogram is a diagram that shows the hierarchical relationships between instances. It is most commonly created as an output from hierarchical clustering. The main use of a dendrogram is to work out the best way to assign instances to clusters.

UBUMonitor helps address educational problems related to personalized learning, collaboration, early intervention, and the effectiveness of teaching strategies. It enables educational researchers to tailor their instructional approaches, improve student outcomes, and contribute to the advancement of educational research and practice. There are many papers that use clustering analysis in education. For example, the paper [5] uses K-means clustering technology to analyse students' global assessment data, with the aim of determining the regularity of involvement of the data, accurately diagnosing learning problems and laying the groundwork for developing effective student management strategies. Other papers, such as [6], analyse the advantages and disadvantages of the various clustering algorithms, evaluate the efficiency and compare the algorithms with educational datasets.

UBUMonitor has been referenced in the literature as an innovative and necessary learning analytic tool [7–11]. The applied use of UBU-Monitor is shown, in a more concrete way, in works such as [12], where the application of extracted log data for heat map generation becomes important for measuring the use of chatbots. Other works, such as [13], apply the tool for dropout detection, through the collected logs. In particular, the clustering module is used for the analysis and classification of the monitored students.

2. Software description

UBUMonitor empowers educational researchers to fully implement a clustering process with educational data (see Fig. 1). Educational data is often multilevel hierarchical and not independent in nature, so an educational researcher must carefully choose the clustering algorithm that justifies the research question in order to obtain valid and reliable results.

2.1. Software architecture

UBUMonitor has been developed as a desktop application with Java and JavaFX as its GUI library. With a hybrid solution, the charts are generated with JavaScript libraries embedded in HTML pages and displayed in JavaFX Webviews.

In the early stages, it was decided to distribute the releases as a Java executable file (extension .jar). Gradually, in later stages, it was distributed as portable and native executable for different platforms such as Win32, Win64, GNU/Linux, and MacOS. Due to user update problems, it was later decided to include a launcher to search for new versions and download them automatically with minimal user interaction. Therefore, the development has been currently divided into two projects:
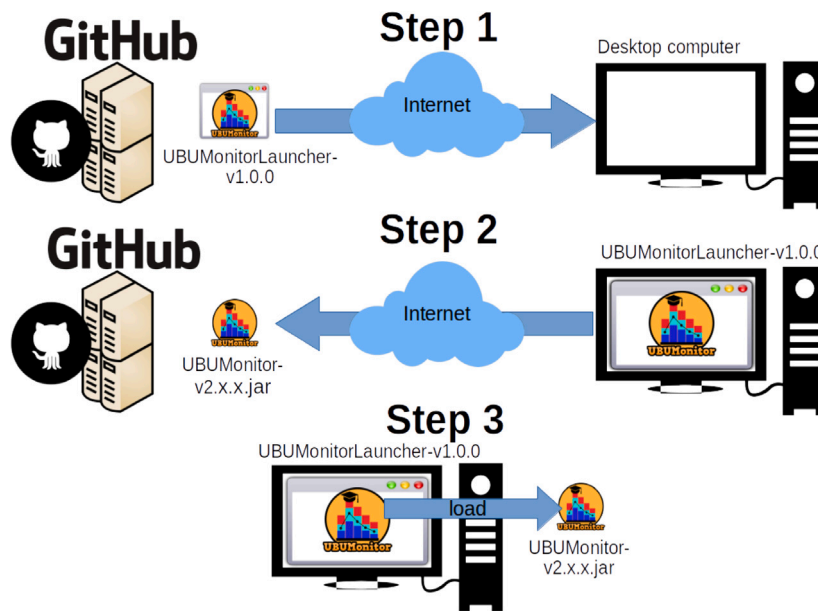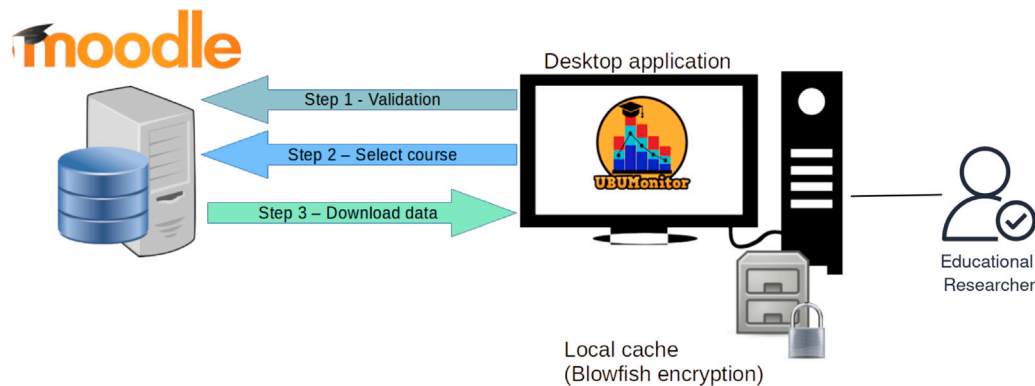
Fig. 2. Deploy Application with Github.

Fig. 3. UBUMonitor's architecture and data download process.

- UBUMonitorLauncher with the exclusive code for checking new versions, downloading them, and launching the application (available at https://github.com/yjx0003/UBUMonitorLauncher/).
- UBUMonitor: with the application code (available at https://github.com/yjx0003/UBUMonitor).

In Fig. 2, the process of deployment in the client desktop computer is shown. First, there is the download of the application launcher (step 1), which checks for and downloads any updates to the application (step 2) before running it (step 3). The first two steps are transparent to the user with the latest locally available version.

UBUMonitor's architecture is easy to use and different in that it only requires an HTTPS/HTTP connection to validate and to select the course, and to download the data. Once the information is downloaded and stored in a local persistent cache, all other interactions can be fulfilled offline without further requests to the Moodle server. Additionally, operating the program in offline mode works without a network connection, loading previous data courses in a transparent way. This architecture and the data download process are shown in Fig. 3.

Moodle data can be downloaded using different techniques. The login is solved using web scraping to obtain the user token, as it is no included as a web service. The course list, gradebook, and activity completion are collected using the REST-API web service available in Moodle 3.9. The CSV file with the logs is downloaded directly via an HTTP/HTTPS request. All HTTP/HTTPS communications are optimized with the okhttp3 API. In order to collect data from the Moodle site, the LMS administrator needs to enable the administrative options that allows access to the Moodle site via Web Service, and also used with the mobile Moodle app. On the other hand, with web services enabled, a teacher only has access to information that can originally be accessed via the web.

Once the data are cleaned and processed, they are stored on the local file system. Very fast subsequent access is possible with this solution, minimizing network requests and time processing. Only if the user wishes to refresh the local data will they have to connect in order to download the latest snapshot once again from the Moodle system.

An important aspect of UBUMonitor is security: all local files are encrypted with the user key by applying the Blowfish algorithm before storage. UBUMonitor allows both HTTPS and HTTP protocols, simplifying the use of remote, local, and intranet servers.

2.2. Software functionalities

Once the educational researcher has validated and chosen the course, they can set up their analysis to create groups of learners based on their interactions and grades with the resources and activities available in the particular course.

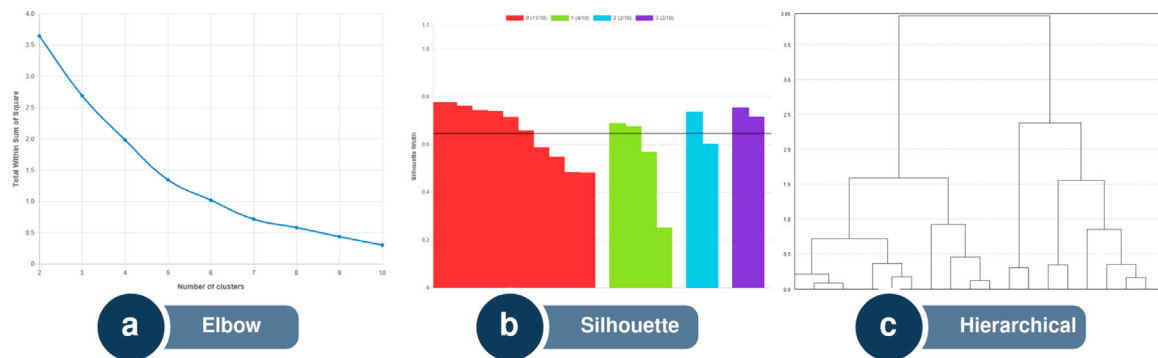The educational data clustering process [3] involves the following steps:

Fig. 4. Example of grouping student log activity with UBUmonitor's clustering functionality. Educational data clustering process is applied [3]. Panels 1a and 3c are image masked to protect student privacy.

1. Data preparation and cleaning. Select the students (instances) and select the course components, their grades or activity completion information (features).
2. Clustering identification. Select a clustering algorithm by specifying the parameters of the number of groups, choosing a distance measure and filtering the features. Run the selected algorithm.
3. Cluster evaluation and validity. Validate the results of the clustering algorithm: searching for the optimal number of clusters, visually checking the cluster properties and checking the assignment of students to groups.
4. Deploy results tagging student groups manually inside Moodle LMS and conducting teaching actions on each group.

3. Illustrative examples

A real use case example is depicted in Fig. 4. This figure shows the steps followed by an educational researcher interested in finding out the interaction profiles of the students in three collaborative activities that they have conducted in the first three weeks of a four-month course. In the two panels on the left, (1a) the 23 students whose learning is to be analysed have been selected, and (1b) three types of collaborative learning activities have been identified in the course (Database, Forum and Glossary). At the top panel, in the clustering tab

(2), the educational researcher has configured and executed a clustering algorithm. In this case the teacher has the possibility to ask the tool (3a) to validate the optimal number of clusters for the selected dataset. In the lower panel, (3b) visualizes and validates the result of the four clusters and (3c) analyses the groupings of the students in the clusters in the table. The educational researcher could notice that there are two minority clusters of two and three students.

With the results of the clustering algorithm the educational researcher identifies three groups of students and can make informed decisions related to early dropout and clustering, known as the deployment task. The teacher identifies the group of students who are not interacting and can send them a message offering tutoring. Another use is to create groups of students for collaborative activities with different behavioural profiles/clusters.

UBUMonitor provides many functionalities to improve the validation process of clustering results: elbow method (a), silhouette coefficient (b) and dendrogram visualization of hierarchical clustering (c) (see Fig. 5). For the graphical representation of 2D and 3D clustering, the PCA algorithm is used to reduce the dimensionality of the features.

Another capability designed to improve the understanding of the results is the management of labels, which allows the number associated with the cluster to be changed to a more meaningful text, e.g. students with difficulties.

Fig. 5. Clustering validation techniques.

4. Impact

UBUMonitor implements the full functionality of the data clustering process. In this way, it empowers educational researchers to utilize the results of student clustering in their decision-making to enhance their teaching methods.

Some new research questions that can be pursued as a result of monitoring student activity on Moodle LMS and analysing clusters: enhancing collaborative learning, optimizing personalized learning paths, evaluating the impact of a new educational activity or resource [3].

UBUMonitor improves the tracking of existing educational research questions. By analysing student activity data, such as which types of resources they access most frequently or which activities they engage in most, the software can help educational researchers group students with similar learning styles. This allows materials and teaching methods to be customized to address different learning styles.

Educational researchers can use UBUMonitor to effectively compare, analyse and reflect on their new interventions to improve collaborative learning and identify the group of students struggling with open software. The UBUMonitor tool is currently included in the official catalogue of online teaching tools at the University of Burgos. Over the past few years, four editions of a course have been conducted to train university staff in using the tool, with a total of 119 attendees. According to the usability surveys conducted with tool users, it received a score of 1.5 on the attractiveness scale (-1 to 2.5), which is higher than the average score of other training tools. Additionally, the surveys show that the tool is clear, efficient, reliable, stimulating, and innovative, scoring above-average scores in each metric. In the best of our knowledge, UBUMonitor has been used in at least 23 subjects across 7 university bachelor degrees and 1 master degree, including computer science, health engineering, education, health sciences, nursing and occupational therapy. It involves data from around 4333 students.

UBUMonitor has a modular architecture, and the clustering module has not been presented before. We describe UBUmonitor's core functionality for monitoring student activity with some educational use cases in these two papers [8,14]. UBUMonitor's code and releases are available on Github. The value of our latest check of the number of downloads of all releases in the Github repositories is 1400.

The nature of both the UBUMonitor software and its development is open source and is available on Github. It was born from a final degree project at the University of Burgos. Its maturity was achieved thanks to internal funding from the same University and a European teaching innovation project.

5. Conclusions

Moodle provides the functionality to group students by specifying parameters such as number of groups or number of students. The criteria for group membership are either random or delegated to the teacher. This is why software that groups students according to their activity and grades in a learning management system is important for the teacher to make informed decisions with different groups/learning styles. A desirable goal of online education is to define personalized learning paths. To achieve this, knowing the maximum number of student groups is a prerequisite.

In order to implement the process of educational data clustering presented in [3], multiple software applications related to data processing and data mining clustering are required. UBUmonitor is a breakthrough because it implements all tasks in the process, in a single desktop application. This functional integration empowers the educational researcher to focus on improving their teaching based on the clusters.

In e-learning platforms such as Moodle, the responsibility for the pedagogical design of courses falls on the educational researchers. UBUMonitor is a cross-platform desktop application that gives educators the possibility to make flexible analysis of how students learn using Moodle. The solution based on a cross-platform desktop application with an installer/launcher as opposed to other alternatives such as web applications frees teachers from advanced administrative knowledge.

CRediT authorship contribution statement

Raúl Marticorena-Sánchez: Conceptualization, Methodology, Software, Supervision, Writing – original draft, Writing – review & editing. **Carlos López-Nozal:** Project administration, Software, Validation, Writing – original draft, Writing – review & editing, Funding acquisition, Investigation. **Ana Serrano-Mamolar:** Investigation, Validation, Writing – original draft, Writing – review & editing. **Alicia Olivares-Gil:** Investigation, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2024.101727.

References

[1] Lang C, Wise AF, Merceron A, Gašević D, Siemens G. What is learning analytics?. In: Lang C, Siemens G, Wise AF, Gašević D, Merceron A, editors. The handbook of learning analytics. 2nd ed. SoLAR; 2022, p. 8–18, Section: 1.

[2] Brooks C, Thompson C. Predictive modelling in teaching and learning. In: Lang C, Siemens G, Wise AF, Gašević D, Merceron A, editors. The handbook of learning analytics. 2nd ed. SoLAR; 2022, p. 29–37, Section: 3.

[3] Dutt A, Ismail MA, Herawan T. A systematic review on educational data mining. IEEE Access 2017;5:15991–6005.

[4] Ezugwu AE, Ikotun AM, Oyelade OO, Abualigah L, Agushaka JO, Eke CI, et al. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Eng Appl Artif Intell 2022;110:104743.

[5] Liu R. Data analysis of educational evaluation using K-means clustering method. Comput Intell Neurosci 2022;2022.

[6] Valarmathy N, Krishnaveni S. Performance evaluation and comparison of clustering algorithms used in educational data mining. Int J Recent Technol Eng (IJRTE) 2019;7(6S5).

[7] Stecula K, Wolniak R. Advantages and disadvantages of E-learning innovations during COVID-19 pandemic in higher education in Poland. J Open Innov Technol Market Complexity 2022;8(3).

[8] Ji YP, Marticorena-Sánchez R, Pardo-Aguilar C, López-Nozal C, Juez-Gil M. Activity and dropout tracking in moodle using ubumonitor application. IEEE Rev Iberoamericana Tecnologias Aprendizaje 2022;17(3):307–17.

[9] Dobashi K, Ho CP, Fulford CP, Grace Lin M-F, Higa C. Learning pattern classification using moodle logs and the visualization of browsing processes by time-series cross-section. Comput Educ Artif Intell 2022;3:100105.

[10] Dobashi K. Sync ratio and cluster heat map for visualizing student engagement. In: Educational data science: essentials, approaches, and tendencies: proactive education based on empirical big data evidence. Springer; 2023, p. 255–89.

[11] Gil-García IC, Fernández-Guillamón A. SMART: Selection model for assessment resources and techniques. Educ Sci 2024;14(1).

[12] Sáiz-Manzanares MC, Marticorena-Sánchez R, Martín-Antón LJ, Díez IG, Almeida L. Perceived satisfaction of university students with the use of chatbots as a tool for self-regulated learning. Heliyon 2023;9(1).

[13] Sáiz-Manzanares MC, Rodríguez-Díez JJ, Díez-Pastor JF, Rodríguez-Arribas S, Marticorena-Sánchez R, Ji YP. Monitoring of student learning in learning management systems: An application of educational data mining techniques. Appl Sci 2021;11(6):2677.

[14] Marticorena-Sánchez R, López-Nozal C, Ji YP, Pardo-Aguilar C, Arnaiz-González Á. UBUMonitor: An open-source desktop application for visual E-learning analysis with moodle. Electronics 2022;11(6).