

Automated Enrichment of Course Structure into Moodle logs: The Contribution of Context-Aware Structural Information to Advance Learning Analytics

DANIELA ROTELLI, Sorbonne Université, CNRS, LIP6, F-75005, Paris, France

YVES NOËL, Sorbonne Université, CNRS, LIP6, F-75005, Paris, France

SÉBASTIEN LALLÉ, Sorbonne Université, CNRS, LIP6, F-75005, Paris, France

VANDA LUENGO, Sorbonne Université, CNRS, LIP6, F-75005, Paris, France

This article presents a methodology for advancing Learning Analytics by automatically reconstructing the organisational structure of Moodle courses leveraging platform logs and backup data. Using an extensive, pseudonymised dataset gathered from twelve undergraduate courses, we illustrate how student activity records can be enriched with structural information. This methodology integrates xAPI statements with parsed XML data from Moodle course backups to create directed graph representations that capture both hierarchical and sequential relationships among sections and activities. By introducing graph-based structural modelling into Learning Analytics practice, we provide new tools to facilitate replicable and context-aware analytics, enabling deeper insights into the relationship between course structure and learning behaviours, as well as actionable enhancement of digital course design.

CCS Concepts: • **Applied computing** → **Learning management systems**; • **Networks** → **Network structure**; • **Information systems** → **Information integration**.

Additional Key Words and Phrases: xAPI statements, Log data, XML structure, Moodle, Directed Graphs, Course structure.

ACM Reference Format:

Daniela Rotelli, Yves Noël, Sébastien Lallé, and Vanda Luengo. 2026. Automated Enrichment of Course Structure into Moodle logs: The Contribution of Context-Aware Structural Information to Advance Learning Analytics. In *LAK26: 16th International Learning Analytics and Knowledge Conference (LAK 2026)*, April 27-May 01, 2026, Bergen, Norway. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3785022.3785119>

1 Introduction

Learning Management Systems (LMSs) play an essential role in the organisation and delivery of blended and online education. Highly customisable and adaptable to individual requirements, these platforms offer a versatile set of tools for supporting teaching and learning. Moodle, one of the most used LMSs with more than 480 million unique users as of September 2025 (*stats.moodle.org*), supports education by providing two main types of learning and teaching modules: *resources*, items that can be browsed online or downloaded; and *activities*, more advanced modules involving student interaction. Moodle also includes the possibility of organising course content through *sections*, which contain resources and activities; *labels*, which serve as a spacer on the home page and enable the integration of text, images, or multimedia between other resources; and *course formats*, pre-structured ways of organising course content into sections to assist

Authors' Contact Information: Daniela Rotelli, daniela.rotelli@lip6.fr, Sorbonne Université, CNRS, LIP6, F-75005, Paris, Paris, France; Yves Noël, yves.noel@lip6.fr, Sorbonne Université, CNRS, LIP6, F-75005, Paris, Paris, France; Sébastien Lallé, sebastien.lalle@lip6.fr, Sorbonne Université, CNRS, LIP6, F-75005, Paris, Paris, France; Vanda Luengo, vanda.luengo@lip6.fr, Sorbonne Université, CNRS, LIP6, F-75005, Paris, Paris, France.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).

Manuscript submitted to ACM

Manuscript submitted to ACM

teachers in designing their courses¹. However, despite the availability of various organisational tools, activities and resources can only and inevitably be arranged sequentially, necessitating vertical scrolling on the page. The presence of links, navigation drawers, and breadcrumbs can facilitate navigation, but comprehending the organisation of a course and retrieving the material can be challenging for learners navigating the content, particularly for extensive courses. It goes without saying that the course content organisation, namely the *course structure*, and its potential structural complexity, could influence the behaviour of learners, thus impacting Learning Analytics (LA).

Current LA research on LMS data commonly concentrates on the granular analysis of student actions [9] and typical analyses focus on similar activities such as participation in forums [1, 8] or quizzes [23]. However, to our knowledge, these analyses do not differentiate activities based on course structure, for example, which section they are located in, their chronological sequence, whether they are always/group accessible or only to a limited extent/group. Our experience indicates that this occurs due to the challenges associated with obtaining course structure information. In Moodle, for example, this information is entirely lacking in the logs, necessitating a laborious manual reconstruction.

Unfortunately, the lack of structural information can affect the investigation of the influence of course structure on the temporal dimension of learning, learning habits, engagement, and results, factors that are particularly relevant in blended and online learning environments, where technology integration can differ significantly from course to course [16]. Additionally, when a structural difference exists between courses, replicating analytical methodologies and comparing learning behaviours might be inconsistent. Does potential inconsistency stem from the analyses being non-replicable and valid solely for a specific dataset, or does it reflect the significant differences among course structures?

To address these challenges, in this paper, we introduce a novel methodology for automatic enrichment of Moodle logs with detailed information on the course structure to enable comprehensive analyses. By enriching xAPI-based logs with data parsed from course backup files, we then represent the structure of courses as directed graphs, capturing both hierarchical and sequential relationships among sections and activities to help characterise different types of course. This supports the replicability of research across courses and enables new forms of structure-aware analytics, laying the foundation for future network-based analyses.

2 Background & Related Work

Courses tend to be organised to align with educational purposes, whether employed as a content repository or as a sequence of educational activities with a defined time frame [15]. LA research focusses extensively on extracting and analysing LMS log data to understand and improve learning and teaching practices. Numerous studies have been conducted on Moodle logs to model student behaviour [26], predict performance [10] and cross-curricular competencies [11], understand collaboration [13] and engagement [7], improve data literacy [2], and determine the temporal dimension of learning [17]. However, to our knowledge, prior work has focused on user interaction with resources and activities without distinguishing the specific organisational context in which they occur.

Recent methodological innovations lay the groundwork for enriching Moodle log data [20, 21], but do not address the automatic reconstruction and integration of full course structures. Emerging advances involve the use of the xAPI (Experience API) standard², which enables the collection of richer and more interoperable event data [22]. Nevertheless, works employing xAPI statements do not connect learning events to a representation of course structure [3, 12, 24]. To date, no studies have explored how structural attributes, such as the location of an activity within a course, or the

¹MoodleDocs - Resources: <https://docs.moodle.org/405/en/Resources>; Activities: <https://docs.moodle.org/405/en/Activities>; Course home: https://docs.moodle.org/405/en/Course_homepage; Text and media area: https://docs.moodle.org/405/en/Text_and_media_area; Course formats: https://docs.moodle.org/405/en/Course_formats.

²Experience API specification - <https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#parttwo>.

interplay between sequential/nested sections and student access, impact learning outcomes, metacognitive processes, and engagement patterns. Nor has previous work offered methods for ensuring the comparability of LA results across potentially structurally heterogeneous courses. This leads to limitations in understanding the learning scenarios envisioned by teachers and, given the potential structural diversity of course designs, hinders the replicability of analytics across different courses or educational contexts. The enrichment of xAPI data with Moodle course backup files, which we propose in this work, is novel in enabling the reconstruction of the entire course as a directed graph, thus automatically enriching the learning traces with comprehensive structural details.

Graph theory [6], the mathematical study of networks consisting of nodes (vertices) and their connections (edges), offers a powerful paradigm for modelling the complex and highly relational organisation of educational content. Graph-based approaches have been extensively used to analyse complex systems such as social networks [25] and computer networks [27]. LA also investigates the social aspects of learning (social integration, social capital, social interactions, community belonging) and the networks that result from them [19]. Social network analysis (SNA) is important in educational sciences and has seen increasing trends in recent years [28]. In the field of LA, its approaches and methods are used to understand the connections that involve actors (students, teachers, courses) in the learning process [4], for example, how connections between students can influence learning performance based on the learning content of the discussions, the frequency of interactions or the classmate with whom to study [5].

Graph representations provide a natural way to model the rich and interconnected structure of online courses. Since courses are inherently structured, that is, modules are grouped into sections, resources and activities are interlinked, and access can depend on sequential or hierarchical organisation, representing such structures as graphs could allow researchers not only to visualise but also to analyse critical attributes such as connectivity, centrality, and navigational pathways within courses [14]. Given the methodological need for structure-aware learning trace enrichment in LA and replicable analytics, this work advances the current state-of-the-art by making such reconstruction and enrichment methods as well as the related code openly available.

3 Educational Context and Dataset

The study was carried out as part of the NLSU project at the Faculty of Science of Sorbonne Université. The NLSU project aims to offer innovative and multidisciplinary bachelor's degree programmes in the fields of science and engineering. Among its goals, the NLSU project aims to create blended courses to ensure good logistics and consolidate support mechanisms. Twelve courses were selected with teachers who agreed to participate in the project for development and delivery in blended format. After data consolidation (Section 4.2), the dataset consists of 2,126,839 actions performed by 2,713 students (Table 1). Users gave their consent to the analytical use of their activity data.

The courses differ in the number of students and teachers, the subject, the semester, the number of ECTS, and the course year: 1st, 2nd, 3rd. In-person classes were generally organised between three and six hours per week of theory and hands-on practice, spanning 19 weeks for the first semester and 20 weeks for the second. Moodle courses, structured to complement lectures and laboratories, were organised with various resources and activities according to teacher's needs. It should be noted that teachers received no initial guidance on how to create a 'blended' course. However, upon request, Sorbonne Université provides teachers with a pedagogical service for the creation and structuring of courses to support teaching activities, as well as the co-design and implementation of teaching training plans³.

³<https://capsule.sorbonne-universite.fr/le-centre/>.

Table 1. Datasets overview of the 12 courses. Courses are ranged according to the year of the bachelor’s degree (1st, 2nd, 3rd). For each course we provide: subject, semester, ECTS (*European Credit Transfer and Accumulation System*), number of students and teachers, students to teacher ratio (#S:t), number of student actions (#St. actions), and number of actions for one student (#A:s).

course	subject	semester	ECTS	#students	#teachers	#S:t	# St. actions	#A:s
1 st								
A	Biology	S2	9	596	55	10.8	677,653	1,137
B	Biology	S1	6	632	59	10.7	598,981	948
C	Physics	S2	6	1,042	77	13.5	462,555	444
D	Environment	S2	3	69	49	1.4	71,840	1,041
E	Business Management	S2	3	33	2	16.5	3,244	98
2 nd								
F	Physics	S1	6	37	4	9.3	36,273	980
G	Informatics	S1	3	32	2	16.0	39,659	1,239
H	Informatics	S1	6	93	4	23.3	107,483	1,156
I	Biology	S1	6	19	2	9.5	3,263	172
J	Physics	S2	6	132	10	13.2	117,404	889
3 rd								
K	English for research	S2	3	16	2	8.0	4,332	271
L	Earth Science	S1	6	12	6	2.0	4,152	346

4 Methodology

Our work began with a detailed study to determine the most suitable technique to collect Moodle data (Section 4.1). In the data pre-processing phase (Section 4.2), we consolidated the dataset by applying the methodology proposed in [20] although adapting it to our dataset, which implied some adjustments. In the data integration phase (Section 4.3), we extracted additional information from the collected xAPI data (CourseID, ObjectID, ItemID, QuestionID), detected fake students when assigning roles, and identified actions performed on deleted activities (Status) and Groups. Lastly, in the data enrichment phase (Section 4.4), we leveraged Moodle course backup files to reconstruct the entire course structure as a directed graph that we used to enrich the dataset with information about the course structure, thus enhancing the context and value of the data. Figure 1, which can be zoomed by clicking on it, depicts the phases of this methodology. Every phase is developed in Python and the entire repository is fully documented and available on GitLab⁴.

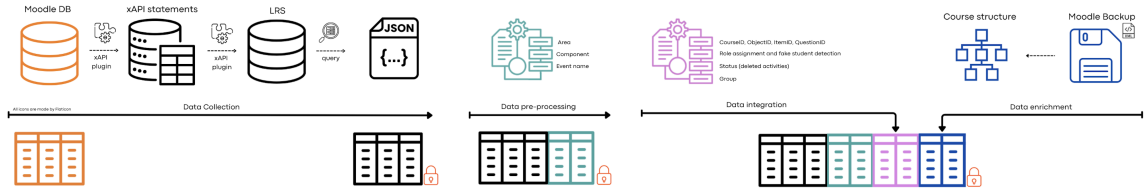


Fig. 1. The phases of the proposed methodology. The padlock represents data where privacy has been respected.

4.1 Data Collection

Moodle stores user activity in a relational database (moodleschema.zoolia.io/). When a user accesses a course, submits an assignment, takes a quiz, or participates in forum discussions, a record of that activity is stored in the Moodle database [21]. Recorded log data can be extracted directly from the database (*database logs*) or from the Moodle log

⁴GitLab Repository - Moodle-xAPI-XML-Integration - <https://gitlab.lip6.fr/mocah-public/moodle-xapi-xml-integration>.

generation interface (*standard logs*), although both are specific to Moodle and suffer from important drawbacks [22]. Another way to collect Moodle data is via the Logstore xAPI plugin [22] (xAPI plugin in the following) which allows to convert Moodle logs into xAPI statements², a standardised data format that enables interoperability among learning platforms [18]. Data are then sent to a Learning Record Store (LRS) where data from other platforms can be integrated in the same format, thus maximising interoperability. Furthermore, the xAPI plugin enables complete pseudonymisation of personal information for data privacy: it mitigates unwanted information breaches by hashing the email address, the user full name, and the user ID using the Sha-1 PHP function (<https://www.php.net/manual/fr/function.sha1.php>) with a secret key before transmission to the LRS. Therefore, we opted for the xAPI format because it is currently the most effective method to gather the most comprehensive information about student actions preserving their privacy.

4.1.1 Moodle logs via xAPI statements. The xAPI plugin was installed in the Moodle (version 4.1) platform of Sorbonne Université. All logs converted into xAPI statements were then sent from the xAPI plugin to the LRS *Learning Locker* of the University to be stored. The data provided by the LRS were stored in a MongoDB database. To extract the data from the database, we requested the IT team to collect data following the method proposed by Rotelli and Monreale [21], who recommend retrieval of all actions performed on the entire platform by each user engaged in the analysis, rather than limiting the focus to actions related to a single course. However, since in their work they do not use the xAPI plugin, we approached the extraction in the following way.

Every xAPI statement is provided with mandatory fields, but can be complemented by a variety of optional fields², as a result, due to the vast amount of data stored in the LRS, the entire dataset from the MongoDB database was not extracted. It was first queried to obtain, for each course (Table 1), the list of users that performed at least one action in the course (to perform an action a user must be enrolled in a course or have a system role (see later Section 4.3 - *Assign roles*). Then, since our goal was to collect all records of the enrolled users in that course, the database was queried to retrieve all the actions that each enrolled user had taken within and outside of the course. For example, if 200 enrolled users in course Z performed at least one action inside the course, all the actions that each of these 200 users performed on the entire platform were extracted.

To access the nested structure of an xAPI statement, it is necessary to specify a query condition on fields of an embedded document (“row” in SQL terminology) using the dot notation, such as ‘statement.object.definition.name.en’ to obtain an object’s name in English. Thus, to retrieve information relevant to learning activities and exclude unnecessary information of the optional fields², the MongoDB database was queried from the following fields:

```
'User': 'statement.actor.account.name',
'Timestamp': 'statement.timestamp',
'Object_id': 'statement.object.id',
'Context': 'statement.object.definition.name.en',
'Description': 'statement.object.definition.description.en',
'RelatedActivities': 'statement.context.contextActivities.grouping',
'Path': 'statement.context.extensions.http://lrs&46;learninglocker&46;net/define/extensions/info.event_name'
```

All data were extracted in JSON format by the IT team in charge of the Moodle platform at Sorbonne Université. The JSON file, which does not contain any message text or user information, is then normalised through the Python json library (<https://docs.python.org/3/library/json.html>) to be visualised as a table.

Privacy protection. When *standard logs* are collected [21], they are first downloaded, then pseudonymised for analysis. To know which users are students, a list of names is required. Consequently, sensitive user information is likely to be

Table 2. Example of a JSON record extracted from the LRS then normalised.

Timestamp	User	Object_id	Context
2023-09-11T18:44:20	d01c53814276c0116335ed14007767e70fb8e	'https://[anonym]/mod/page/view.php?id=22711'	Course questions on variables
Description	Path	RelatedActivities	
Consider the following instruction: >> a==b; Tick the statement(s) that are TRUE.	\mod_quiz\event\attempt_submitted	'https://[anonym]/course/view.php?id=153', 'https://[anonym]/mod/quiz/view.php?id=22957', 'https://[anonym]/mod/quiz/attempt.php?attempt=39993&cmid=22957'	

accessible throughout these stages. In addition, if two students share the same name (Moodle distinguishes users based on their ID, which is not a field of *standard logs* [21]), we could assign actions to a user who did not performed them. Moodle xAPI statements are fully pseudonymised. User information was hashed using a secret key exclusively known to the IT team, thus allowing the protection of user privacy and data sharing among the researchers. Moreover, as we will explain in Section 4.2, we did not need the list of students to retrieve their logs. A sample record is shown in Table 2.

4.2 Data Pre-processing

For the data pre-processing phase, we followed the methodology proposed by Rotelli and Monreale [20]. However, the authors did not use the xAPI plugin, which required some adjustments. Standard logs are supplied with nine features: *Time* (yyyy-MM-dd 'T' HH:mm), *User full name* (the user performing the action), *Affected user* (potential recipient of the action), *Event context* (within the platform), *Component* (learning module types), *Event name* (types of action performed on the component), *Description* (description of the event), *Origin* (such as web or web service), and *IP address*. Due to a series of pitfalls that Moodle hides, in their work, the authors redefine and align **components** and **event names** and integrate data with the **area** of the platform (within and outside courses) in which the action occurs.

Information about the component and the event name is absent in our data. Therefore, we first had to retrieve this information leveraging the feature *Path*. For example, in Table 2, the path \mod_quiz\event\attempt_submitted indicates that the event name *attempt_submitted* is performed on the component *mod_quiz*. Once we retrieved this information, we could apply their methodology for data pre-processing.

4.3 Data integration

Role assignment and fake students detection. In Moodle, a role is a collection of permissions that can be assigned to specific users in specific contexts⁵. When users log into the platform, the 'authenticated user' role is assigned. System roles, such as administrator, manager, and course creator, are assigned at the system level, namely throughout the entire platform, including the front page and all courses. Users can be assigned the role of teacher, non-editing teacher, or student only within a course. A user can have multiple roles on the platform, in different courses, or in the same course. For example, a user can be a teacher in one course and a non-editing teacher in another. Moreover, a user can be both a teacher and a student in the same course (typically for testing activities). Since our data are pseudonymised, we do not know which users are students, teachers, or managers. As a consequence, we proceeded as follows.

Whenever a user is enrolled in a course, two events are recorded: 'User enrolled in course' and 'Role assigned'. The same happens when a user is unenrolled from a course ('User unenrolled from course', 'Role unassigned'). Since each

⁵MoodleDocs - Roles - https://docs.moodle.org/405/en/Roles_and_permissions.

role has a specific ID (in our Moodle installation: 1: manager, 5: editing teacher, 7: non-editing teacher 9: student)⁶, the analysis of the sequence of role assignments and unassignments, along with the role ID specified in the 'Object_id' field (Table 2), enabled us to understand the course **Role** of each user.

The above may seem trivial. However, as previously mentioned, there may be cases where managers and teachers assign themselves the role of student in order to test some student-side functionalities or to take a look at the various courses. Now consider courses with thousands of students where enrolment can take place throughout the year, as can unenrolment (change of course, non-payment of fees, withdrawal from studies). Having an up-to-date list of enrolled students is not always easy and may also not be privacy-compliant. Hence, we would run the risk to retrieve the logs of potential *fake students*. The bias resulting from the analysis of this data would be significant.

We highlighted that for data collection (Section 4.1), we took into account not only the actions performed by a user in a specific course, but all actions performed on the platform. For example, if user 1 has performed at least one action in course Z, we take into account any actions performed by user 1 across the entire platform, which includes all courses in which user 1 has performed other actions. This means that we can identify if user 1 was enrolled in other courses and, most importantly, the role they have been assigned in these courses.

This enabled us to detect *fake students* in two ways. Firstly, we detected users who were assigned an additional role other than Student in the same course⁷. For example, user 1 in course Z was assigned the role of Student and Editing Teacher. Secondly, we detected users who were assigned the role of Student in a course but in other courses were assigned a role other than Student. For example, user 2 in course Z was designated as a Student, in course W as a Manager, and in course Y as an Editing Teacher. In the first case, we determined the role according to the hierarchy defined by capabilities: *i) Manager, ii) Editing Teacher, iii) Non-editing Teacher, iv) Student*⁸. Therefore, in course Z, user 1 was exclusively assigned the role of Editing Teacher, and that of Student was removed. Similarly, it may happen that by mistake a manager assigns themselves the role of teacher instead of manager when enrolling in a course. They were exclusively assigned the role of Manager. Meanwhile, in the second case, a Student enrolled with a hierarchically higher role in other courses was assigned the role of 'Other courses' to easily identify them. In course Z, user 2 was assigned the role of 'Other courses' in course Z, as they are neither a Teacher nor a Student in the course, but rather a participant enrolled in other courses who likely visited course Z for observation. In our dataset, we detected 171 fake students, which enabled us to remove them from our investigations, thus preventing incorrect results.

CourseID, ObjectID, ItemID, QuestionID. Actions can be performed on the platform or in a specific course. A student can access two different forums with the same name or submit many files for the same assignments until the deadline. The 'RelatedActivities' field (Table 2) stores information about the context of an action in a nested way. For example, if a student accesses a course, this field will store information about the course (CourseID). Similarly, if a student accesses a module, this field will contain information about the course and the module (ObjectID). Going deeper, if a student answers a question, the field will store information about the course (CourseID), the quiz (ObjectID), the attempt (ItemID), and the specific question (QuestionID). We therefore extracted this information, which misses in Moodle standard logs, to integrate the dataset with the following attributes **CourseID, ObjectID, ItemID, QuestionID**.

Status. Teachers can delete modules and their related data (e.g., grades, files) from Moodle. Similarly, students can delete forum posts, comments, Wiki pages, choices, submissions, etc. Logs of actions performed on deleted modules remain accessible in the Moodle log list, as these actions occurred prior to deletion. Also *Quiz attempt* events can be deleted.

⁶The IDs can change in different Moodle installations. The complete list of roles is available via *Site administration > Users > Permissions > Define roles*.

⁷Please note that on Moodle a student can be assigned a teacher role for specific activities. As a result, we considered only the course context, not the activity context: https://docs.moodle.org/405/en/Assign_roles.

⁸Moodle Docs - Standard roles - https://docs.moodle.org/405/en/Standard_roles.

For example, a teacher is preparing a quiz but overlooks marking it as unavailable, resulting in some students taking it. Adding or deleting questions can be done, provided that there are no attempts of the quiz. Therefore, potential taken attempts must be deleted. However, students have taken them. In the reference paper for data pre-processing [20], the authors removed activities performed on deleted modules since “considering this type of learning activities would be worthless”. We partially agree with the authors. In fact, despite the deletion of the modules or attempts, the actions were performed and this information is highly relevant for sequential, process, or temporal analysis. Consequently, we resolved to identify them for a case-by-case evaluation about their filtration or retention. The ‘Description’ field allows to identify deleted modules or events via the *deleted* value, which we categorise as “DELETED” in the field **Status**. Events or modules that are not deleted are “Available”.

Groups. Moodle enables teachers to create groups. Groups are extremely helpful in courses with multiple teachers, each managing their own group of students. Groups are also useful for assigning a particular activity, resource, or topic section to a single group, preventing another group from accessing it, or separated by **Groups**. To identify the group to which a user was added, we proceeded in the same way as for the role. We leveraged two fields (Table 2): ‘Event name’ (Group member added/removed) to identify the added/removed users and ‘Context’ to identify the name of group. It is important to note that students can be added in more than one group. Therefore, we created a **Group** field to contain the list of groups in which a student was added.

4.4 Data enrichment

Although the integrated attributes thus far are highly beneficial to extend the scope of analyses, xAPI statements do not provide information about the course structure to enable a comprehensive overview. In an attempt to provide more context and to make data more actionable, thus increasing the quality of knowledge that can be derived from it, we took a step forward and enriched the dataset through the information we derived from the backup of the course.

A Moodle course can be saved through the **Course backup** from the course navigation menu by selecting *More > Course reuse > Backup*. Teachers, without requiring admin capabilities, can generate a backup of their courses (or parts of them by selecting specific sections or activities) for storage or reuse. A backup file is characterised by the MBZ extension and its name follows the format *backup-moodle2-courseid-coursename-date-hour.mbz*, ending with *-nu.mbz* for backups without user data and *-an.mbz* when the option ‘Anonymize user information’ is checked. After unzipping the backup file, it is made up of a list of XML files and folders containing additional folders and XML files. As illustrated in Figure 2a, there are four folders: activities, course, files, and sections.

Activities. This folder contains a set of folders representing all the resources and activities modules of the course¹. Each folder in turn is named according to the format *moduleType_courseModuleId*. For example, ‘assign_2817’ (Figure 2b) represents the assignment with course module id (cmid) 2817. Please note that cmids are sequentially assigned in the entire platform according to the datetime of creation. Hence missing numbers do not represent deleted modules, but modules stored in other courses. The folder of each module contains in turn a set of XML files that depends on the module type (Figure 2c). In this work, for each module, we took into consideration the *module.xml* file and the *module_type.xml* file (e.g., *asssing.xml*, *page.xml*, *quiz.xml*).

Course. This folder contains several XML files and may include additional folders (such as blocks) based on the availability of optional components in the course. We focused our attention on the *course.xml* file (Figure 2d) that contains information such as the course id, name, format, startdate, and enddate. For completeness, it is worth noting that the file *enrollments.xml* contains information about the users who have been enrolled in the course, while the file

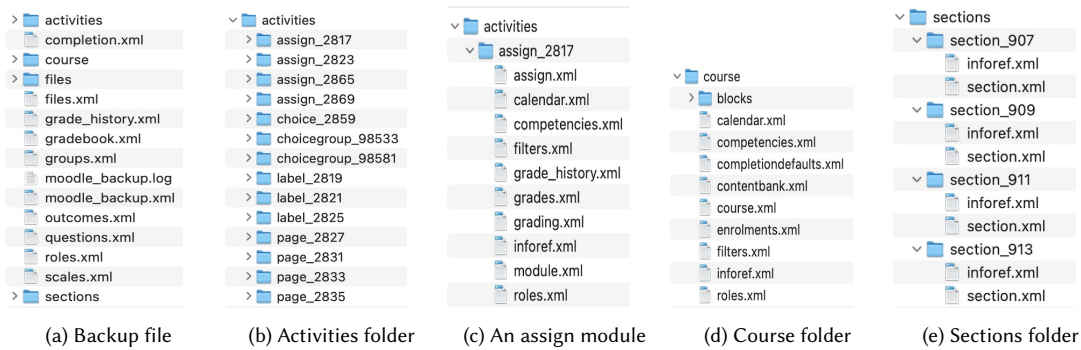


Fig. 2. Exploded view of the contents of a backup (XML/MBZ) file

roles.xml instructs about the role each user has in the course. Since our data are pseudonymised (Section 4.1) and our backup files lack user data (-nu.bmz), these two files do not contain any user information, therefore preserving user privacy. The *roles.xml* file in the backup file (Figure 2a) includes the list of role types used in the course.

Files. This folder comprises a list of enumerated subfolders, each storing the encrypted versions of files such as PDFs, videos, images, spreadsheets, etc.

Sections. Similarly to activities, this folder includes a set of subfolders whose name follows the format *section_sectionId* (Figure 2e). The *inforef.xml* file indicates which files are contained in a specific section and in what order. The *section.xml* file provides information on the section number, name, description (if any), whether it is visible (a section, like a module, can be hidden from students), the cmids in the order in which they appear, and any access restrictions. Depending on the course format (Section 1), additional information can be stored in this file. For example, in the *Flexible sections* and *Grid* formats the tags `<name>` and `<value>` allow reconstructing the nested structure of the course.

Upon parsing the Moodle backup, we successfully reconstructed the entire course structure as a directed graph using the Python library *Networkx*. We started from the *course*, the root, connected by edges to the *Section* nodes. Each section includes activity nodes and potentially sub-section nodes. We proceeded until we exhausted all *Activities* and *Sections* within the backup file (Figures 2b and 2e), reaching the leaves. Each node contains a set of attributes: its name, its categorisation as a section or activity, and the associated settings. For example, since the settings encompass visibility parameters (certain course elements can be hidden from students), we also provided this information. In this way, we were able to enrich the dataset with details regarding the structure of the course, namely, for each action we identified the section (or subsection) in which each action was performed, along with its settings.

It is important to note that the Logstore xAPI plugin is updated by developers as new versions of Moodle are released. The current backup format has been used since *Moodle 2.0* (2010) and the current Moodle log format has been used since *Moodle 2.6* (2013). Therefore, neither the Moodle version nor the plugin version nor the backup version affect the proposed methodology in any way. For this reason, we have not benchmarked against the versions.

5 Results and Discussion

To showcase the value of the proposed methodology, we applied it to the 12 Moodle courses described in Section 3. We report first on the actual content of the enriched records as made available by our methodology (Section 5.1); then

we provide the graphs of the hierarchical course structures and we report on similarities and differences based on our observation as experts; finally, we illustrate how the enriched records combined with the graphs allow to better understand learners' access to activities and resources on Moodle (Section 5.2).

5.1 Automatically enriched traces with course structure information

Figure 3 shows samples from the enriched dataset, which result from the methodology presented in Section 4. Each colour corresponds to a different phase as illustrated in Figure 1: *i)* black indicates the information that we extracted via the xAPI plugin: User, Timestamp, and Context; *ii)* green corresponds to fields that we pre-processed according to the methodology of the reference paper [20] although arranged to our data: the Area of the platform (course, authentication, profile, etc.), Component (the learning module type), Event name (the action performed on the module); *iii)* purple illustrates the fields that have been integrated using the information retrieved from the XAPI statements: the ObjectID, the ItemID, the QuestionID, the Status of action (available or deleted), the Role, the Group (in case of students). Please note that the *Description* field is also present in traditional Moodle logs as the description of the action performed (e.g., user X posted some content in forum Z). However, xAPI statements provide additional information: if the action is performed on a question, the 'Description' field returns the text of the question, as in the example; *iv)* blue indicates the fields that allowed us to enrich the dataset with information about the structure extracted from the Moodle backup: the module's Visibility status (false if hidden to students), the Availability (access restrictions is any), the Section names, and the Section number to provide exact structural identification.

Moodle logs			Data pre-processing			xAPI statements					Moodle backup					
User	Timestamp	Context	Area	Component	Event_name	Description	ObjectID	ItemID	QuestionID	Status	Role	Group	Visibility	Availability	Section_name	Section_number
98a0d3aed6ce202a1abd880	2023-09-12T16:10:24	The forum	Course	Forum	Some content has been posted	the forum assessable	51363	727	-	Available	Teacher	-	True	-	General	0
f28aabc7b97c9f641613b04dbf	2023-09-23T10:41:09	Course questions on iteration TP 5: Spyder II (For further reading)	Course	Quiz	Quiz attempt submitted	What is the value displayed at the end of execution [list=[10,100,1000,10000]]-0 while i == 2: i=i+1 print(i) list2=[]]	44171	81841	316239	Available	Student	A	True	-	Le cours	13
f6ca668a559898e2ced1b98f58	2023-11-13T14:58:30		Course	Assignment	A file has been uploaded	the assignment assessable	82109	-	-	Available	Student	B	True	-	Le TP/ID	20
						Indicate where Python passes through in the following example. NB: We assume that the INSTRUCTIONS do not modify the value of x. x = 6 if x < 10: x = 7 : INSTRUCTION_1 else : INSTRUCTION_2 INSTRUCTION_3								Student must match all of the following: Activity 33876 must be complete with pass grade		
8509d9b1ac0cf611c0834b0d8	2023-09-27T11:07:04	Course questions on conditions	Course	Quiz	Quiz attempt submitted		50989	97053	395945	Available	Student	B	True		Le cours	10

Fig. 3. An example of enriched records (click to enlarge).

We now discuss the value of the specific fields extracted with our methodology and what type of analytics they could enable. Consider the opportunity to get a comprehensive perspective not only of the quiz (*ObjectID*) the student is undertaking, but also of the specific attempt (*ItemID*) they are engaging in and (the same quiz can be taken many times), and for that attempt, the specific question (*QuestionID*) they are dealing with, which is available in the *Description* field. The analysis may focus on the specific questions answered by students in each attempt, the frequency of quiz repetitions by certain students, the questions that took the most time, among other factors. In the context of a forum, it would be possible to identify which discussions the student explored, in which they created content, and on which they invested the most time. A sequential examination of interactions inside the forum discussions could be conducted without direct access to them: the xAPI plugin enables the retrieval of all conversations while protecting student privacy.

Additionally, the *Visibility* may be essential to comprehend, for example, the changing patterns over time of student behaviour with respect to specific activities and resources. What is the reason that specific activities/resources have been used exclusively by particular students? What factors contribute to certain students declining to revisit previously engaged activities or resources? If an activity or resource is visible, it is possible to examine the reasons students engage with or avoid it. If an activity was previously visible and is now hidden, students cannot access it again. This unveils

hitherto unexamined domains for study. Another crucial factor to consider is the *Availability*. What factors contributed to the completion of this activity by only certain students, while others did not participate? What was the reason for this activity to be conducted only within a specific timeframe? Understanding access restrictions, based on group, date, grade, or completion of previous activities, provides analytical opportunities that were previously unimaginable.

The *Group* field can complement *Availability* when constraints pertain to the group and facilitate analysis by group. The use of groups can be useful to identify groups of students with better performance or greater difficulties, or with differentiated learning strategies that favour a certain path, or groups that use some available resources but not others.

Section_name specifies the section's name, and *Section_number* has been introduced to provide exact structural identification, as two sections may share the same name. The course structure is dependent upon the teacher's organisation, specifically regarding the structure of activities in terms of time and organisation. Consider how analyses may be differentiated not just by activity, but also by section. Are there sections that students seldom or infrequently access? Are they not highly identifiable? Do they have visibility and availability parameters, and are these parameters restricted to some groups or by date? Are the sections arranged sequentially, or do they collect an entire set of similar activities (e.g., all quizzes, all PDFs, all videos)? Analyses grounded in structural organisation, situated within the realm of teaching analytics, can unveil novel research possibilities.

Answering these questions was only partially feasible by accessing the course to verify the parameters and manually integrating them. While ensuring student privacy, our methodology can now be used to automatically enrich the traces with information on the structure of the course, taking the first step towards a comprehensive analysis and investigation of the student's complex behaviour.

5.2 Graph representation of courses

Course organisation depends on choices that can vary from teacher to teacher and that might include a different use of the LMS (e.g., content repository, assessment tool, student interactions, collaborative work) [15], but also the presence of subsubjects or student subgroups, and even teacher personal preferences. As previously mentioned (Section 1), despite the availability of various organisational tools to help teachers structure their courses, activities and resources can only and inevitably be arranged sequentially, necessitating vertical scrolling on the page. At first glance, courses all seem to be organised in the same way. Links, navigation drawers, and breadcrumbs can help navigate a course, but comprehending the organisation of a course and retrieving material can still be difficult, thus impacting LA.

5.2.1 Graph representation. Starting from the representation of the course as a directed graph, where nodes represent sections or activities, and edges represent organisational relationships, we aimed to visualise the hierarchical tree of the course structured into sections containing activities, resources, and potential sub-sections. Hence, we used two functions of the Networkx library: `dfs_tree` to return nodes in increasing order according to their section number, and `graphviz_layout` to create the node positions for the graph. In this way, we transitioned from a sequential perspective, where courses containing sections, activities, and resources, were largely analogous, to a hierarchy-based view where the structure of the course can be clearly identified. This also enable mapping the section number of the enriched records to the corresponding node in the tree. Figure 4 shows the hierarchical tree of six courses. The display is reduced due to space constraints, but the image can be enlarged and downloaded for detailed viewing. Representations of all structures are available in the GitHub repository⁴. Upon observation, we noticed that some structures had remarkable visual similarities; additionally, we also analysed the titles of the sections and the activities contained in each section.

Courses G and H (Figures 4c and 4d) feature very similar hierarchical structures and visual organisations. In both courses, branches extend to multiple levels: we can detect sections, sub-sections, and even sub-subsections (in orange) showing a parent-child relationship between course components. Both courses also present two sections for course management (General, Information/Introduction) containing informative activities such as forums, URLs, and pages. If we look at the section titles, in both cases we can notice an ascending numbering (1, 2, 3, etc.). Since the coloured nodes help to distinguish between content types at a glance, for each section (or subsection), we can observe the same sequence of learning activities and resources. In course G, the pattern is represented by a page followed by a quiz. In course H, for each section, we first encounter a subsection collecting URLs, then a subsection collecting quizzes, followed by a subsection of files. We regularly encounter assignments (Sections 6, 8, 9 for course G; Sections 3, 7, 9 for course H) coloured red. The last section is usually different from the rest of the sections and contains a series of

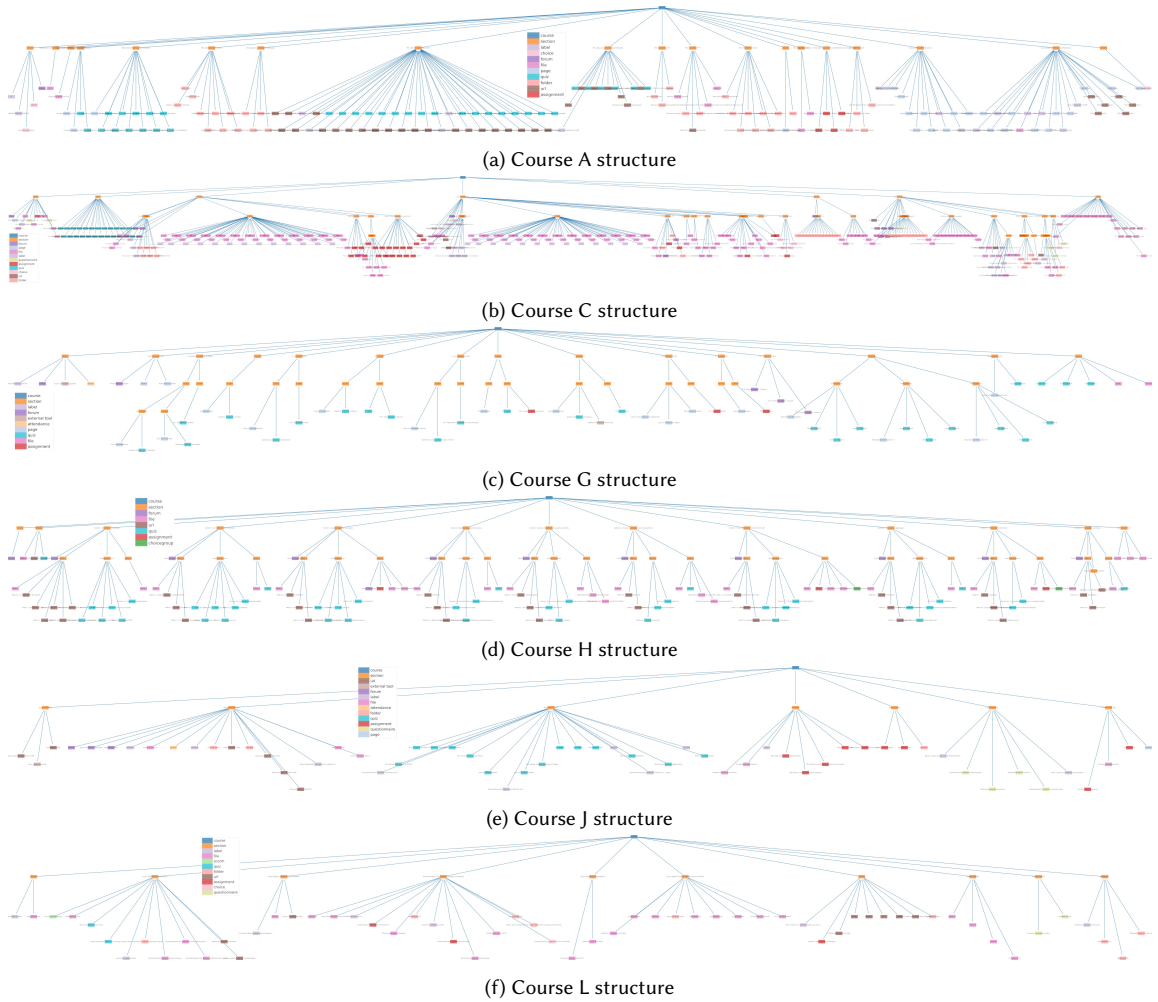


Fig. 4. The structure of courses with nodes representing sections or activities and edges representing organisational relationships.

quizzes and files for course G, and only files for course H. The sequential nature of the titles, with recurring activities interspersed with assignments, makes it easy to follow the flow, the temporal distribution, the density, and dependencies of the course content. This led us to classify these two courses as structured by *chapters*.

Looking at courses J and L (Figures 4e and 4f), they are organised in a very similar way. Both courses have a section for course management (Information/General) that contains informative activities for the course. Both courses have a section that groups resources and activities with a common functionality: in course J, Section ‘outils numériques’ (digital tools) contains a series of links to digital tools external to the platform that are used during the course; in course L Section ‘Introduction à la minéralogie’ (introduction to mineralogy) contains SCORMs, files, quizzes, URLs, and folders used for the introductive functionality. Observing the titles, the sections are organised according to the tasks that students need to complete. In course J we can identify: *i*) Auto-positionnement (self-assessment) which collects several self-assessment tests to check mastery of a topic; *ii*) Mini-projets that students need to complete independently; *iii*) Projet en équipe, for group projects. In course L we can identify: *i*) Travaux pratiques (practical work) that refer to teaching sessions in which students carry out practical work, experiments, or projects to apply theoretical knowledge in a practical way and develop practical skills; *ii*) Travaux dirigés (tutorials), namely small group classes where students apply theoretical knowledge, work on exercises, ask questions, and interact with the teacher to deepen their understanding; *iii*) devoirs (homework). With respect to courses structured by *chapters*, courses structured by *tasks*, namely organised with sections that contain as a set of resources and activities that enable a specific task to be completed, do not present a regular pattern in the activities within each section and do not have any noticeable temporal sequence. The insights we can glean from the structure suggests that access to resources and activities is fairly unrestricted and based on self-regulation or instructions provided by teachers from time to time.

In courses A and C (Figures 4a and 4b) we can recognise a common structure that we identify as *activity types/tasks*. In these courses, the sections are organised in the same way as in the task-based courses. For example, there are sections for “travaux pratiques/dirigés” (practical work/guided work) and “auto-positionnement” (self-assessment). Compared to task-based courses, courses A and C also have sections that exclusively contain the same type of activity. It is clear from looking at the images that some sections only contain activities of the same colour. For example, in course C there is a section that only contains quizzes, while in course A some sections only contain folders. The title analysis then confirmed the hypotheses: course C section with only quizzes is called “Quiz”; course A section with only folders is called “Mes diaporamas” (my slides). Based on our expertise in blended learning and in Moodle, we classified courses D, E, G, H, I, K, as organised by *chapters*; courses F, J, L by *tasks*; courses A, B, C by *activity types/tasks*. Some courses, such as course C, are divided into subgroups (students, teachers, tutors) before being organised by task and activity.

5.2.2 Students’ access analysis. In light of this preliminary examination, which prepares the way for future investigations into the effect of a course’s structure on students’ behaviour, we wondered if there was a data-driven way to confirm our manual categorisation. In courses structured by chapters, does the sequential nature of the structure affect access? In courses organised by tasks, do the periods of time when a task must be performed imply a higher density of activity? Can we analytically identify the structure of a course based on student access to course material?

Thanks to the enriched records, we are now able to illustrate the density of student engagement with course materials over weeks, along with the course structure, as depicted in Figure 5. Since the amount of actions can vary between sections based on the number of activities and resources within each section, which imply varying levels of student actions, we used the function `normalize` ([sklearn.preprocessing](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html)) to normalise data per section.

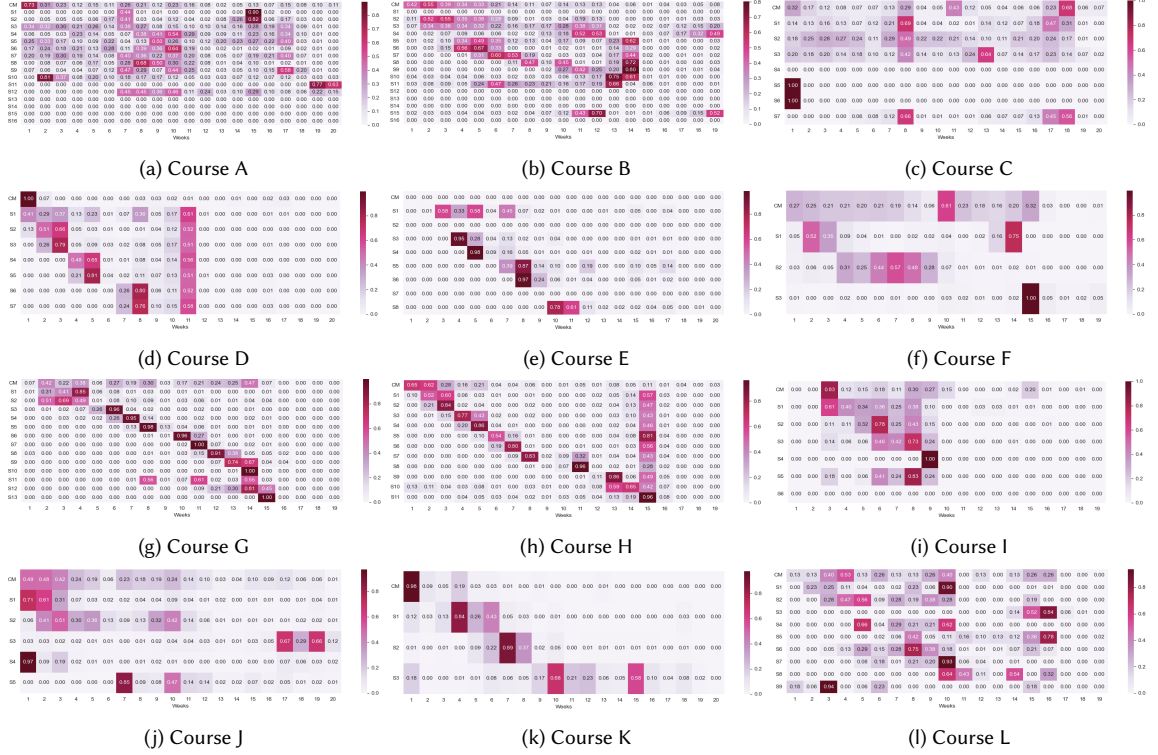


Fig. 5. Student access density to course material over semester weeks (click to enlarge).

Although the total number of sections varies, reflecting variations in structure or granularity between courses, we identified similarities and differences between the heatmaps. Please note that some heatmaps show entire sections without access, that is, with cells equal to 0 throughout all weeks. This indicates that the sections are hidden, that access is restricted to teachers only, or that the sections are empty (in the latter case usually at the end of the course).

In courses D, E, G, H, and K, most heatmaps show that the activity of “Course Management” (CM) peaks mainly in the first weeks, indicating a stage of onboarding or course setup. Course E has no activity in the CM section because this section is missing in the course. It is important to stress that sections show a clear trend of access density moving sequentially over time. Students appear to concentrate their activity in one section (e.g., S1) for a block of weeks, then transition to the next section (S2), and so forth. This left-to-right movement of high-density blocks suggests that students follow the course structure in an organised sequence, accessing each new section as it becomes relevant or available, with minimal overlap between sections. Although course I reveals a somewhat distinct access density, we can still identify sequential peaks. Being the briefest course in terms of duration (the course appears to have spanned merely six weeks), the activity may have been highly concentrated. This data-driven classification completely reflects our manual classification by *chapters*. It is worth noting that courses organised by chapters end much earlier than others, indicating that students are almost guided through their studies and, once the course is over, they no longer access it.

In courses F and J, CM section shows activity spread throughout the course duration, reflecting a consistent access to general resources, communication tools, or coordinative elements across all weeks. Upon examination, we observed

that the CM section of course J contains the URLs to distance learning sessions, whereas the peak experienced in course F in week 10 corresponds to the "assignment" activity, likely an examination located in the CM for immediate retrieval. With respect to courses organised by chapters, these two courses have more diffuse peaks, showing more distributed activity throughout weeks and sections, suggesting a *task*-based course design. In fact, activity drops significantly after week 11/12 and then resumes 4/5 weeks later. Based on the academic calendar, the break corresponds to the period when activities are suspended for the Christmas and Easter holidays. We manually classified courses F, J, and L as organised by task. However, in course L, we cannot identify a clear interruption. The data indicate that course L should be classified alongside courses organised by *activity/task*, which we have identified in courses A, B and C.

In courses organised by *activity/task*, the access to CM section is high during the first weeks and around mid-to-late semester, suggesting central coordination points that drive student activity. Upon examining the course C's peak in week 18, we observed considerable participation in a questionnaire, likely a course evaluation located in the CM section for quick identification by students. Compared to courses organised by task, two aspects immediately caught our attention. First, we cannot observe a clear interruption in activities during the holiday break. Although reduced, activity on the platform remains continuous throughout the semester. Secondly, and more importantly, heatmaps show notable overlapping peaks typically occurring in the middle of the semester. This synchronous timing of peak access in some sections could reflect not only task-based sections, but also sections collecting similar activities (all quizzes, all slides, all videos) that are accessed for self-assessment or self-study to complement the tasks.

6 Conclusion, limitations, and future work

In this work, we introduce a methodological innovation for automatically enriching Moodle logs with course structure information by integrating xAPI statements with data extracted from Moodle backup files to capture hierarchical and sequential relationships between sections and activities. We then identified different structural course patterns based on student density access to learning materials. The proposed enrichment methodology aims to advance the synergy between learning data and graph theory to provide more context and increase the quality of knowledge that can be derived from it. This is meant to enable more comprehensive and replicable analyses of learning behaviours across courses, highlighting the practical impact of graph modelling in educational data analysis. Graph theory applied to education can enable new forms of network analysis supporting not only visualisation but also formal analysis.

In future work, we intend to leverage the enriched records analyse which sections or activities act as hubs or dead-ends, identify structural features that facilitate or hinder learning engagement, quantify the diversity and complexity of possible student paths within a course, explore learning pathways and habits to model student behaviour, calculate the shortest navigational paths for learners to retrieve information, analyse the overall navigability of the course structure, and compare the properties of different courses to improve course designs. In short, we want to investigate how different structural course patterns impact student access to learning materialsto support teaching and learning.

However, despite the novelty of this contribution, it requires the availability of Moodle backups and xAPI logging, which may not be feasible in all deployments. Moreover, despite being supported by data, the evidence reflected in the course structures relies on experts' observations, which may not capture all nuances and differences in course design. The results are also limited by the specific educational context from which data were collected (Faculty of Sciences), which potentially limits the generalisability to other subjects.

It is important to emphasise that although the analysis carried out in this study focusses on Moodle, the enrichment of the dataset with information on structure, reconstruction of course structure to ensure replicability, and the impact

of structure on student behaviour can be performed on any type of LMS. This is only a first step towards the application of course structure in graph-theoretical methods in LA, and much work remains for future research in the field.

References

- [1] Meriem Adraoui, Asmaâ Retbi, Mohammed Khalidi Idrissi, and Samir Bennani. 2017. Social learning analytics to describe the learners' interaction in online discussion forum in Moodle. In *2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET)*.
- [2] Daniel Amo-Filva, Belén Donate Beby, Francisco José García-Peñalvo, and Jiahui Chen. 2022. Towards an ethical data literacy proficiency: a Moodle logs analytical tool. In *2022 XII International Conference on Virtual Campus (JICV)*. 1–3.
- [3] Dimitri Bigler, Julian Manz, Kevin Lee, David Fischer, and Georg Hagel. 2024. Learning Environment Interoperability in Software Engineering Education. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*. 1–5.
- [4] Brian V Carolan. 2013. *Social network analysis and education: Theory, methods & applications*. Sage Publications.
- [5] Maarten De Laat, Vic Lally, Lasse Lipponen, and Robert-Jan Simons. 2007. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for Social Network Analysis. *International Journal of Computer-Supported Collaborative Learning*.
- [6] Reinhard Diestel. 2025. *Graph theory*. Vol. 173. Springer Nature.
- [7] Konomu DOBASHI, Catherine P FULFORD, Meng-Fen Grace Lin, et al. 2019. A heat map generation to visualize engagement in classes using Moodle learning logs. In *2019 4th international conference on information technology (IncIT)*. IEEE, 138–143.
- [8] Aysu Ezen-Can, Kristy Elizabeth Boyer, Shaun Kellogg, and Sherry Booth. 2015. Unsupervised modeling for understanding MOOC discussion forums: a learning analytics approach. In *Proceedings of the fifth international conference on learning analytics and knowledge*. 146–150.
- [9] Sithara HPW Gamage, Jennifer R Ayres, and Monica B Behrend. 2022. A systematic review on trends in using Moodle for teaching and learning. *International journal of STEM education* 9, 1 (2022), 9.
- [10] Ángel Hernández-García, Carlos Cuenca-Enrique, Laura Del-Río-Carazo, and Santiago Iglesias-Pradas. 2024. Exploring the relationship between LMS interactions and academic performance: a learning cycle approach. *Computers in Human Behavior* 155 (2024), 108183.
- [11] Santiago Iglesias-Pradas, Carmen Ruiz-de Azcárate, and Ángel F Agudo-Peregrina. 2015. Assessing the suitability of student interactions from Moodle data logs as predictors of cross-curricular competencies. *Computers in Human Behavior* 47 (2015), 81–89.
- [12] Qais Kalyana Lintang, Niken Dwi Wahyu Cahyani, and Muhammad Arief Nugroho. 2024. Log Analyzer for English Proficiency Test Cheating Detection. In *2024 12th International Conference on Information and Communication Technology (ICICT)*. 115–122.
- [13] Jingyan Lu and Nancy Wai Ying Law. 2012. Understanding collaborative learning behavior from Moodle log data. *Interactive Learning Environments* 20, 5 (2012), 451–466.
- [14] Abdul Majeed and Ibtisam Rauf. 2020. Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions* 5, 1 (2020), 10.
- [15] Jacek Marciniak. 2014. Building E-learning Content Repositories to Support Content Reusability. *Int. J. Emerg. Technol. Learn.* 9, 3 (2014), 45–52.
- [16] Margie Martyn. 2003. The hybrid online model: Good practice. *Educause quarterly* 26, 1 (2003), 18–23.
- [17] Aleksandra Maslennikova, Daniela Rotelli, and Anna Monreale. 2023. Session-Based Time-Window Identification in Virtual Learning Environments. *Journal of Learning Analytics* 10, 3 (2023), 7–27.
- [18] Azer Nouira, Lilia Cheniti-Belcadhi, and Rafik Braham. 2018. An enhanced xAPI data model supporting assessment analytics. *Procedia Computer Science* (2018).
- [19] Oleksandra Poquet and Srecko Joksimovic. 2022. Cacophony of networks in learning analytics. *Handbook of learning analytics* (2022), 38–45.
- [20] Daniela Rotelli and Anna Monreale. 2022. Time-on-task estimation by data-driven outlier detection based on learning activities. In *LAK22: 12th International Learning Analytics and Knowledge Conference*. 336–346.
- [21] Daniela Rotelli and Anna Monreale. 2023. Processing and understanding Moodle log data and their temporal dimension. *Journal of Learning Analytics* 10, 2 (2023), 126–141.
- [22] Daniela Rotelli, Yves Noël, Sébastien Lallé, Vanda Luengo, and David Pesce. 2023. A moodle plugin for rich xapi data logging. In *European Conference on Technology Enhanced Learning*. Springer, 748–754.
- [23] Sandra Schön, Philipp Leitner, Martin Ebner, Sarah Edelsbrunner, and Katharina Hohla. 2021. Quiz feedback in massive open online courses from the perspective of learning analytics: role of first quiz attempts. In *International Conference on Interactive Collaborative Learning*. Springer, 972–983.
- [24] Xenia Stein, Katerina Tsarava, Armin Fabian, Merlin Carl, Alla Kutkina, and Walther Paravicini. 2025. Development and pilot study of an asynchronous online enrichment course on mathematical proving. *Int. Journal of Mathematical Education in Science and Technology* (2025), 1–35.
- [25] Shazia Tabassum, Fabiola SF Pereira, Sofia Fernandes, and João Gama. 2018. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 5 (2018), e1256.
- [26] Feng Hsu Wang. 2017. An exploration of online behaviour engagement and achievement in flipped classroom supported by learning management system. *Computers & Education* 114 (2017), 79–91.
- [27] Peter Whittle. 2007. *Networks: Optimisation and evolution*. Vol. 21. Cambridge University Press.
- [28] Akça Okan Yüksel and Sibel Somyürek. 2023. The use of social network analysis in educational sciences studies. *Journal of Educational Technology and Online Learning* 6, 4 (2023), 1128–1145.