edX Log Data Analysis Made Easy

Introducing ELAT: an open-source, privacy-aware and browser-based edX log data analysis tool

Manuel Valle Torre, Esther Tan and Claudia Hauff
Delft University of Technology
Delft, The Netherlands
{m.valletorre,e.b.k.tan,c.hauff}@tudelft.nl

ABSTRACT

Massive Open Online Courses (MOOCs), delivered on platforms such as edX and Coursera, have led to a surge in large-scale learning research. MOOC platforms gather a continuous stream of learner traces, which can amount to several Gigabytes per MOOC, that learning analytics researchers use to conduct exploratory analyses as well as to evaluate deployed interventions. edX has proven to be a popular platform for such experiments, as the data each MOOC generates is easily accessible to the institution running the MOOC. One of the issues researchers face is the preprocessing, cleaning and formatting of those large-scale learner traces. It is a tedious process that requires considerable computational skills. To reduce this burden, a number of tools have been proposed and released with the aim of simplifying this process. Those tools though still have a significant setup cost (requiring the setup of a server), are already out-of-date or require already preprocessed data as a starting point. In contrast, in this paper we introduce ELAT, the **e**dX **L**og file **A**nalysis **T**ool, which is a browser-based (i.e. no setup costs), keeps the data local (i.e., no server is necessary and the privacy-sensitive learner data is not send anywhere) and takes edX data dumps as input. ELAT does not only process the raw data, but also generates semantically meaningful units (learner sessions instead of just click events) that are visualized in various ways (learning paths, forum participation, video watching sequences). We report on two evaluations we conducted: (i) a technological evaluation and a (ii) user study with potential end users of ELAT. ELAT is open-source; for review purposes we anonymized it and made a short demonstration video available at https://vimeo.com/user103400556/elatdemo.

ACM Reference format:
Manuel Valle Torre, Esther Tan and Claudia Hauff. 2019. edX Log Data Analysis Made Easy. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 10 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Educational research into Massive Open Online Courses (MOOCs) has taken off in recent years, as—among others—evident in the creation of a whole conference series[1] dedicated to it. In contrast to early predictions of MOOCs as being a "revolution" of higher and life-long education, reality has proven to be more complex. The retention rates in MOOCs remain very low [27] and interventions designed to improve MOOC learners' outcomes are often failing when deployed on MOOC platforms with thousands of learners [3]. This is most often the case when topic-agnostic MOOC platforms such as edX and Coursera are investigated, while platforms specifically designed for a particular topic (e.g. language learning platforms, programming platforms) tend to lead to more positive results [3].

Despite these problems, there are distinct advantages to exploring learning phenomena in MOOCs: they are large-scale (often with tens of thousands of learners), platforms typically log all possible interactions the learners have with the material, many MOOCs have been released (and re-released) over time, they attract participants from a wide range of backgrounds (and thus insights are not limited to a particular type of cohort) and they often allow researchers to deploy their own interventions such as an interactive learning planner [5], a webcam-based attention detector [29], a next-step recommender [21] or a collaborative chat [7].

Besides intervention-based research, a large number of studies have also been dedicated to the post-hoc analysis of MOOC learner behaviours, e.g., [2, 10, 12, 16, 37]. What both the intervention-based and posthoc analyses-based works have in common is their reliance on the data traces logged by MOOC platforms in their quest to analyze learners' behaviour. We here focus on the *log traces produced by the edX platform*, as the data each MOOC generates is easily accessible to the institution running the MOOC (and this in turn has led to a lot of learning research being conducted on edX). Those log traces, though detailed, are typically not at the *semantic* level necessary to answer a particular research question. As a concrete example, consider the edX log entry in Figure 1: it contains a single click event (in this case, video pausing) of a single learner being active in a single course. Many of these log entries need to be aggregated to even compute basic statistics such as the number of seconds a learner watched a particular video. This typically requires the writing of a number of scripts (often in Python or R), in order to extract the desired information from the logs. This is not only repetitive and inefficient (as researchers duplicate their efforts to clean, preprocess and aggregate the logs), it also severely limits the insights researchers without a computational background can gain from it, as they have to rely on the precomputed statistics provided by the edX Insights[2] platform.

A number of prior works have started to tackle this issue. In 2014, Veeramachaneni et al. [35] introduced MOOCdb, a shared data model

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] The *ACM Learning @ Scale* conference series was started in 2014.

[2] https://insights.edx.org/

```
{"username": "USER", "event_type": "pause_video", "ip": "000.000.000.000", "agent": "Mozilla/5.0 (Windows
NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36", "host":
"courses.edx.org", "session": "4098c9df34a6668e5a352951ceee407d", "referer":
"https://courses.edx.org/courses/InsituteX/EX2015/courseware/a79124bd10c042cc886fcac3b6f09ee7/9e5e8017/",
"accept_language": "en-US;q=0.6, en;q=0.4", "event": "{\"id\":\"i4x-InstituteX-EX-video-ea41fb\",\"currentTime\"
:46.44,\"code\":\"H_RWt3rxrWE\"}", "event_source": "browser", "context": {"user_id": USER_ID, "org_id": "InsituteX",
"course_id": "InsituteX/EX/1T2015", "path": "/event"}, "time": "2015-04-21T19:29:42.524601+00:00", "page":
"https://courses.edx.org/courses/InsituteX/EX/1T2015/courseware/a79124bd10c042cc886fcac3b6f09ee7/9e5e8017/"}
```

Figure 1: Example of one edX log entry (slightly simplified): a user pauses a video.

which revolves around the types of interactions (observing, submitting, collaborating and feedback) learners can have on a MOOC platform. The accompanying Python implementation converts log traces into a relational schema that can be queried through SQL. A year later, moocRP [20] was introduced, which incorporates data models such as the MOOCdb one and offers a number of out-of-the-box visualization and analytics functionalities, and thus also caters to researchers without programming experience. At the same time, though, the moocRP implementation was last updated four years ago and requires a number of installation steps that are not trivial to execute, especially considering the age of the implementation.

As an answer to these challenges, we introduce ELAT, a browser-based tool that (i) requires zero installation efforts beyond that of a modern web browser such as Google Chrome, (ii) is privacy-aware in the sense that all computations happen on the user's local machine and no data is send to the cloud, and (iii) incorporates a number of analytics and visualization modules (based on feedback we acquired from end user interviews) that can be arranged into a personalized dashboard. ELAT is written in JavaScript, builds upon MOOCdb's data model and makes exclusive use of the modern web browser's functionalities, such as a built-in database, to access functionalities that formerly required a client-server architecture. Longevity of our software is ensured due to the browser vendors' careful browser updating policies—browser updates are designed to *not* break existing web applications.

Having implemented ELAT, we conducted two evaluations—a system evaluation and a user study with seven participants, each of them a potential end user of our tool. We find that even large MOOCs (65K learners) can be processed in a reasonable amount of time (less than six hours) on a standard laptop. Our user evaluation showed that main purposes of ELAT—to remove two major burdens for the learning researcher (to setup tooling and to preprocess the data) and to provide meaningful insights of learner behaviour at various levels of granularity—have been achieved.

In the remainder of this paper, we first present related literature and tooling (§2) and then turn to ELAT and the description of its architecture and capabilities (§3). In §4 we outline our evaluation—both from the technical perspective *(is a standard browser able to handle Gigabytes of log data?)* and the user perspective *(is the tool useful to our targeted end users?)*. Finally, we conclude with an outlook into ELAT's future in §5.

2 BACKGROUND

In recent years, research on MOOCs has witnessed a proliferation of empirical studies on learner attrition rate [6, 9, 11], learner engagement patterns [10, 14, 19, 23, 26, 36, 38] and self-regulation [4, 13, 17]. The above-mentioned research foci often demand more than simple statistical analyses of survey or self-report measures to draw conclusions on learners' online learning experience and learning engagement. In addition, they may also require daily updates to determine for instance whether or not to extend or deactivate an intervention deployed in an A/B test setup [28].

As a result, researchers have been leveraging different tools for log analysis for years now, but it appears to remain an obscure and repetitive process. Often, there is little information beyond *"All log parsing was performed using standard modules in Python and R."* [32] which makes it impossible to fully reproduce the findings as low-level log events have to be aggregated into semantically meaningful units (e.g. at what amount of inactivity does a new learner session start? at what level of activity is a learner considered to be an active MOOC learner—is one visit to the course enough?). This issue has been recognized and addressed by a number of prior works, the most important and relevant ones for us—i.e., open-source tooling that processes edX log files—are listed in Table 1: apart from our own tool ELAT, we consider MOOCdb[3] (one of the first efforts to build process log data into a session-based data model, which they defined), moocRP[4] (a data processing and visualization tool similar in spirit to ELAT with emphasis on server-side security measures), visMOOC[5] (initially a video analysis tool to inspect the aggregated learner activities on course videos but later extended to include forum behavior [8], dropout analysis visualizations [1], and more [6]), ANALYSE[7] (developed as an OpenEdX plugin to address its lack of analytics [30]) and edx2bigquery[8] (tool to import edX log files into Google's BigQuery, a web service for the interactive analysis of large datasets).

We set up our comparison along three dimensions: (i) the tool itself, (ii) the data that goes in and comes out, and, (iii) the available visualizations of learners' behavioural data. For all tools, we first determined whether they can still be installed according to the available instructions—not surprisingly, for the tools whose last software update was a few years ago (MOOCdb, ANALYSE and moocRP) this is no longer possible, due to outdated software libraries[9]. ELAT, even if not developed any further, we expect it to remain working for years, due to the browser vendors' policies of not releasing new

[3] https://github.com/MOOCdb/Translation_software

[4] https://github.com/CAHLR/moocRP

[5] https://github.com/HKUST-VISLab/vismooc

[6] https://elearning.hkustvis.org/

[7] https://github.com/jruiperezv/ANALYSE

[8] https://github.com/mitodl/edx2bigquery

[9] For the subsequent categories of tools that are no longer working, we relied on the tools' available documentation and demonstration videos to gauge their visualization and data capabilities.

Table 1: Overview of edX-based data processing and visualization tools. The ✓ sign indicates "Yes", the ✗ a "No"/"None". In the customization row, ○/◐ indicate minimal/medium amounts of dashboard customization. The last row lists the visualization types: either interactive visualization (VIS) or code samples to showcase how to generate visualizations (SAM).

	ELAT	MOOCdb [34]	visMOOC [33]	ANALYSE [24]	moocRP [22]	edx2bigquery [18]
MOOC Platform	edX	edX & others‡	edX	Open edX	edX & others‡	edX
Tool						
Working	✓	✗	✓	✗	✗	✓
Open Source	✓	✓	✓	✓	✓	✓
Last Software Update	09/2019	06/2015	04/2018	07/2015	11/2015	08/2018
Development Platform	JavaScript	Python, Matlab	Python, TypeScript	Python	Python, JavaScript	Python, BigQuery
User Platform	browser	Python + SQL	browser	browser	browser	Python/R + SQL
Knowledge Required for Setup	✗	Docker, Python, MongoDB, MySQL	Shell, Python, MongoDB, MySQL	Open edX developer, Django Server	Node, MySQL, Redis, Sails	Python, mySQL, BigQuery
Data						
Output (downloadable)	sessions	sessions	✗	events	events	events & sessions
Storage	local	server	server	server	server	cloud (BigQuery)
Incremental Update	manual	manual	✗	automatic	automatic	automatic
Visualizations						
Customization	◐	✗	○	◐	○	✗
Downloadable	✓	✗	✗	✗	✗	✗
Knowledge Required for Operation	✗	Python, R	✗	✗	✗	Python, R
Types	8 VIS, 2 SAM	3 SAM	3 VIS	20 VIS	4 VIS	7 SAM

‡We note that besides edX, multiple data models (including e.g. Coursera) are listed on the respective website, but we found no information to this effect in the accessible code.

browser version that break existing web applications. While for most tools the user platform (i.e., how the end user accesses the tool) is indeed the browser, for all but ELAT (which requires no setup) significant computational knowledge is required in order to set up the tooling as they all employ a traditional client-server architecture. As a concrete example, consider the setup requirements of moocRP: knowledge of Node.js, mySQL, Redis and Sails is required.

In terms of data, most tools allow the downloading of the aggregated data, with ELAT, MOOCdb and edx2bigquery aggregating the logs on a learner session level, instead of an individual event level. In terms of storage, only ELAT stores the data locally (i.e., inside the end user's browser), whereas all other tools require the use of an additional server or the cloud (edx2bigquery). The use of a service such as the Google BigQuery cloud seems tempting, as all processing costs reside in the cloud; however, not only does it require a substantial pipelining effort for an institution to move its data to the cloud, it also leads to issues with respect to data privacy. Apart from visMOOC for which no information is available, all tools allow an incremental update of the logs—as edX logs are released in 24 hour cycles, it would be wasteful to process all data again. Due to ELAT residing completely in the browser, new edX log files have to be uploaded manually; in contrast, ANALYSE, moocRP and edx2bigquery provide an integration with Amazon S3 (the cloud service edX uses to store and serve all edX data[10]) and thus any new log file added to the correct S3 folder is automatically being processed.

Lastly, let us consider the visualization category: while all tools have at least some visualization capabilities (either by providing visualizations out of the box or by providing sample code of how to generate visualizations from the data), for edx2bigquery and MOOCdb accessing those capabilities requires at least some knowledge of programming languages. ELAT allows the downloading of the generated charts and plots for further processing by the end user. Customization of the visualizations is possible for all but visMOOC and moocRP. Finally, we note that ELAT includes eight interactive visualizations while ANALYSE offers by far the most visualizations, but as already stated is no longer being updated.

Overall, we argue that in this comparison ELAT's strengths lie in the complete lack of setup costs, the local processing of log files into semantically meaningful learner sessions and the provision of a range of visualizations based on recent research works.

3 ELAT

In this section, we first outline our design requirements, describe the edX log traces and then turn to ELAT's system architecture. Lastly, we showcase the analytics and visualization functionalities ELAT incorporates.

3.1 Design Requirements

In response to the challenges outlined so far, we have developed ELAT. It is a browser-based application that provides a host of information extracted from edX log traces. After an initial interview with six stakeholders from our local institution who are all involved in the edX MOOC content creation process, we identified a number of requirements:

(1) The tool can be used on different operating systems.
(2) It requires minimal or no installation efforts.
(3) It requires no data preprocessing of the edX log traces.
(4) It requires no configuration.

[10] https://edx.readthedocs.io/projects/devdata/en/latest/access/download.html

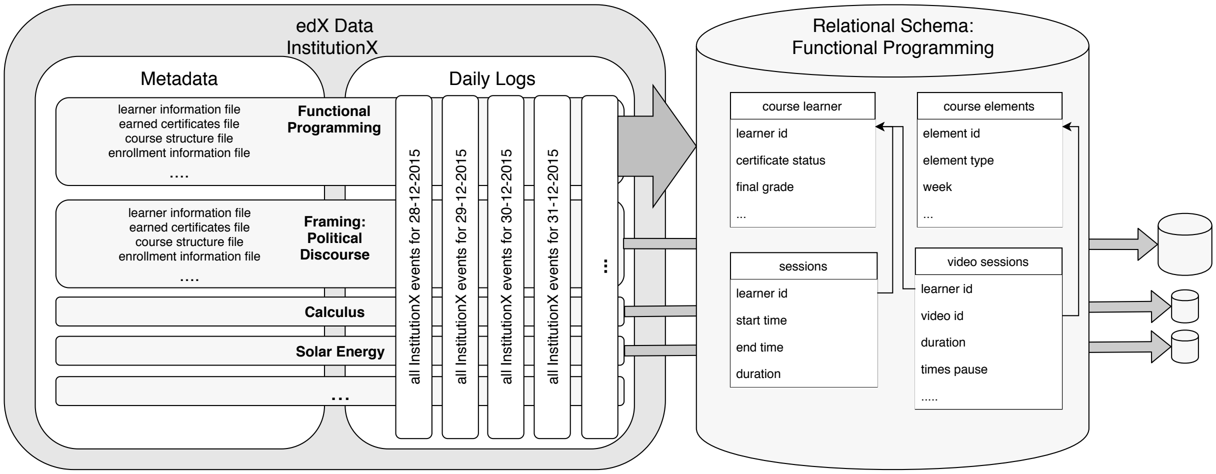


Figure 2: edX data traces are converted to a relational schema which is based on MOOCdb. edX log files aggregate all clickstream data generated on a single day across all running MOOCs offered by an institution. The relational schema is generated for each course separately.

(5) No data leaves the end user's machine as course data contains sensitive information including learners' names, email addresses and so on.

(6) It supports the visualization of controlled trials—especially important when interventions are deployed in an A/B test setup to gauge whether they have an effect.

(7) Aggregated data and visualizations can be downloaded easily for further processing.

(8) As edX releases the daily log files in 24 hour cycles, it should be possible to simply add another log file to an already existing dataset without having to restart the processing pipeline from scratch.

Due to requirements (1) and (2) we settled on the modern web browser as runtime environment for our tool; to the end user, ELAT has the look and feel of a regular web application (though in contrast to most web applications, it will be running all data processing on the client). Initially this may seem like an odd choice, as we are dealing with potentially Gigabytes of course data as seen a number of edX MOOC examples in Table 2: e.g., one run of the self-paced *Solar Energy* MOOC attracted more than 20,000 learners (6% of those successfully completed the MOOC, a percentage in line with expectations [27]). In that time period, more than 845,000 learning sessions[11] were recorded, yielding more than 5 Gigabytes of processed data. Due to requirement (5) we could not develop a traditional client-server application (where data is send to a server that does the heavy lifting in terms of processing), and thus had to rely on the modern browser's web APIs to enable a similar processing pipeline within the browser. All modern browsers support IndexedDB, a low-level web API that enables client-side (i.e., *inside the browser*) storing of large amounts of structured data. As already noted, another advantage of a web-browser based tool is the fact that web browser are designed to run code that was created many years ago, quite a contrast to the multitude of tools and frameworks shown in Table 1 where a single outdated library or a breaking change in a framework can mean a tool to no longer work (unless significant effort is expended to resolve library/framework issues).

Before describing the system architecture of ELAT, let us briefly discuss the makeup of the edX data traces and how to translate those to a semantically meaningful data model (i.e., MOOCdb's data model in our case).

3.2 From Low-level Logs to Relational Schemas

Every edX course is defined by two types of data traces: *metadata* files and *clickstream* files as visible in Figure 2. The former contains information on the individual learners as well as the course makeup. The latter is a click-by-click *log* of learners' activities across *all courses* running at a particular institution *on a particular day*. For the *Solar Energy* MOOC (cf. Table 2) this means that ELAT had to handle the processing of 731 files (that amounts to 34.5 GB compressed files) in order to extract the clickstream data relevant for that particular MOOC.

As the MOOCdb data model provides meaningful semantic units (including forum interactions, forum sessions, survey responses, quiz questions and so on) we decided to use it as a basis for ELAT. Given that the original MOOCdb Python implementation[12] is by now five years old, we started off with a modified version of it[13] that had been updated to be usable with a more recent version of Python.

Python is not natively supported in the browser, a fact that can be remedied in two ways: we can either make use of compilers that compile Python code into JavaScript in an "offline step" (e.g., Transcrypt[14]) or we employ Python interpreters that themselves are written in JavaScript (e.g., Brython[15]). We experimented with both Transcrypt and Brython and found the latter not suitable for our purposes as (i) we cannot access the JavaScript code that is generated on the fly and thus cannot manually optimize it, and

[11] A learning session starts when a learner enters the MOOC and ends after 30 minutes of inactivity.

[12] https://github.com/MOOCdb/Translation_software

[13] https://github.com/AngusGLChen/DelftX-Daily-Database

[14] https://www.transcrypt.org/

[15] https://github.com/brython-dev/brython

(ii) the unoptimized version takes too long to process even small log files (17 minutes for a file of 10 Megabytes—our optimized code now processes the same file in less than 6 seconds). Using Transcrypt, we were able to first generate the JavaScript code, analyze its function and performance, and optimize it manually to fix any issues and increase the speed of conversion. We note that an extensive manual code review was required (approx. 100 hours overall), as Transcrypt's automatically translated code could not be executed as-is due to the fact that many Python to JavaScript data structure conversions had issues[16].

3.3 System Architecture

With the JavaScript code in place to convert data traces to a relational schema, we can now discuss the overall architecture of ELAT, which is depicted in Figure 3. The user interface component enables the user to load the metadata (which determines for which course the clickstream logs are extracted) and clickstream files into the browser's storage. The FileReader module is responsible for parsing and extracting the data in a efficient manner, even at Gigabyte sizes. For the compressed log files (in gzip), we employ the pako[17] library, a zlib port to JavaScript.

Once the data has been read, it is now processed by the Processing module which translates the log files into MOOCdb's data model as already outlined in Section 3.2. The relational data is moved to the browser's persistent storage, that is, IndexedDB. As IndexedDB does not support fixed-column tables (like a relational database) out-of-the-box, we employ JsStore[18] as a relational wrapper for IndexedDB and the accompanying SqlWeb as a SQL query interpreter. This in turn allows us to formulate SQL queries to extract the desired learner information and aggregates.

As our runtime environment is the browser, we also need to discuss storage limits. There is the *global limit* the browser has, which is half of all available disk space on the local machine. Secondly, there is also the *group limit* for websites (to the browser, ELAT is just a website) of the same origin or domain—this limit is 20% of the global limit. As an example, if a machine has 500 GB of free disk space, the browser can use 250 GB for persistent storage (global limit) and 50 GB (group limit) for a single origin or domain such as ELAT.

Once the database instance has been set up, the Dashboard component is responsible for rendering a number of charts. The popular visualization libraries apexcharts.js[19], Chart.js[20], and D3[21] are employed to populate the user's dashboard.

Lastly, the Downloader component is responsible for processing the data into csv format, which our end users can use as input to another program (R, SPSS, etc.). For instance, it is possible to plot the distribution of video watching sessions using the video interactions csv file, and even go a step further and use the course learner file to compare the plot of certified and uncertified learners. This sample is included in ELAT's documentation [22].

3.4 User Interface Choices

The user interface consists of two parts: (i) the tabular overview (Figure 4) which provides basic information extracted from the metadata and log files and (ii) eight visualizations, which provide more specific insights and were chosen based on the recent literature and our interview with our stakeholders.

Due to space constraints, we here only discuss three of the eight visualizations (those based on recent research works):

- The learning path visualization (Figure 5) [2] provides an overview of the paths learners take through the different *types* of MOOC components. Each edX MOOC has a limited number of components (video, forum submit, quiz start, quiz submit and so on) and the transition probabilities from one to the next are computed based on all learner traces (though this can also be filtered according to successful/non-successful learners). In Figure 5, for instance, we observe that nearly 40% of learners transition to the forum once they have ended their quiz (presumably because they have questions and are looking for answers).
- The forum analysis chart (Figure 6) shows different types of forum posters (regular posters, regular forum readers, etc.) and how their numbers develop across the course weeks [25]. We find forum posts by regular posters to peek midway, while most occasional posters have contributed posts in the first weeks of the MOOC.
- The video watching sequence visualization (Figure 7) [2] presents information to what extent learners that pass the course follow the prescribed video watching learning paths. Each dot represents a video, colored by course unit, and the layout from left to right is according to the designed (i.e., instructor-provided) learning path. The width of the edges indicate the percentage of learners moving from one node (i.e., video) to the next.

The visualizations themselves are interactive and users can use mouse hovers, or select different segments, *such as passing or failing learners*, and date ranges, to receive more information on the different plots. In addition, as for the table view it is possible to filter learners according to certain criteria, such as separating learners into segments by their edX id (this is useful if interventions were randomly deployed to learners based on their edX ids). This allows learning researcher to for instance find out whether the inclusion of an intervention led to a higher passing rate.

4 EVALUATION

Having described ELAT's goals and design, we now describe the two types of evaluations we conducted: a technical evaluation (§4.1) and a user study with potential end users (§4.2).

4.1 System Evaluation

In order to investigate to what extent ELAT can handle MOOC data, we employed ELAT on four edX MOOCs (cf. Table 2)—we selected

[16] For instance: to check if an element belongs to a set, the command in Python is `if element in set`; Transcrypt translated the code into JavaScript as `if set.includes(element)`. This translation works for a JavaScript array but the correct notation for a JavaScript set is `if set.has(element)`.

[17] https://github.com/nodeca/pako

[18] https://jsstore.net/

[19] https://github.com/apexcharts/apexcharts.js

[20] https://github.com/chartjs/Chart.js

[21] https://github.com/d3/d3

[22] Hidden for anonymity

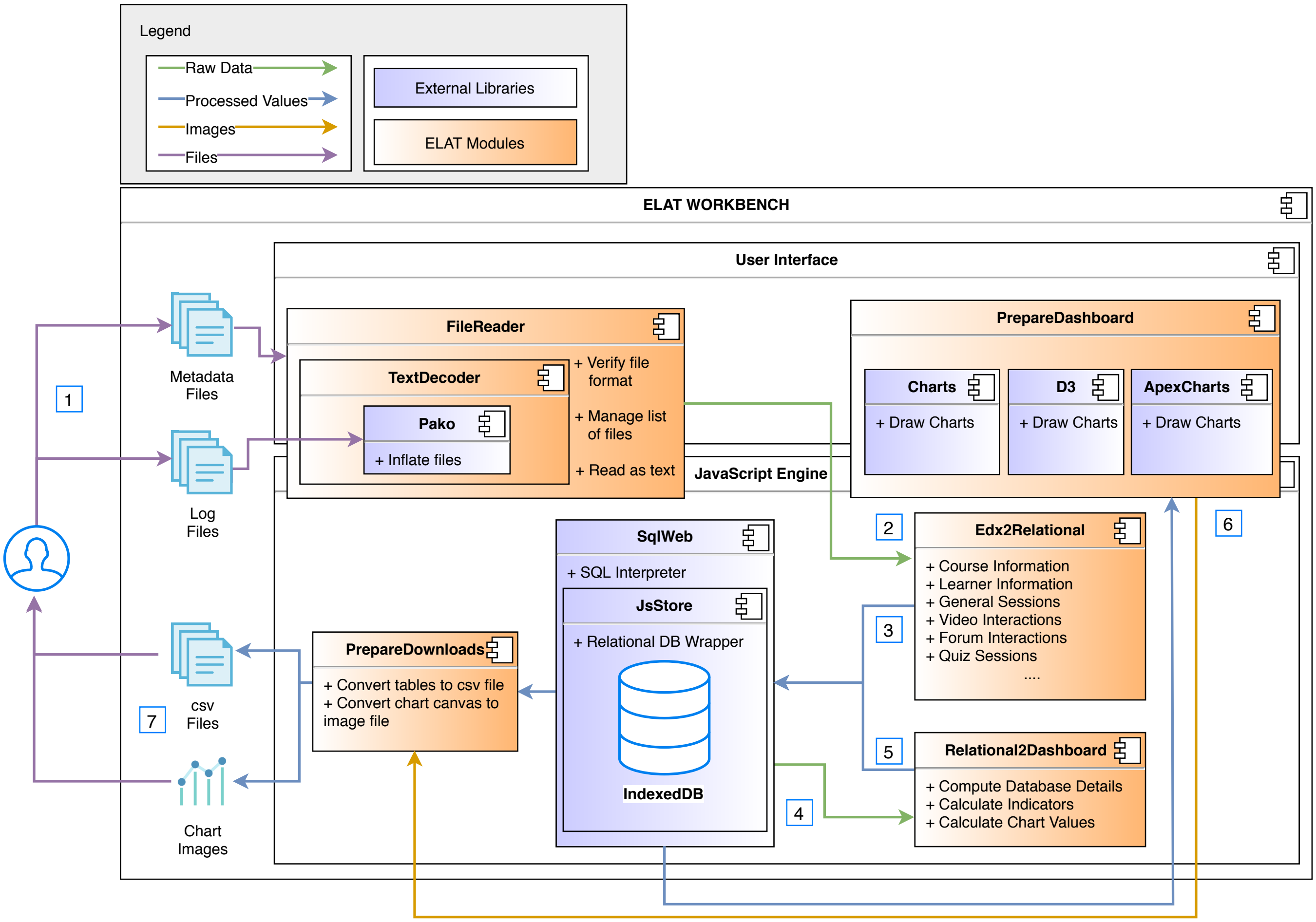


Figure 3: ELAT Architecture

those due to their different learner sizes (between 2.7K and 65K learners), the different years they ran (between 2015 and 2019) and the vastly different time interval they were open for enrollment (between 83 and 731 days). Recall that edX logs are organized by day, so the size of the log file depends on the number of MOOCs running concurrently by the institution. On average, the log file size is 50 Megabytes compressed and more than 1 Gigabyte uncompressed, with hundreds of thousands of click records.

4.1.1 Processing Times & Disk Space. We evaluated the processing time on two different machines with the most recent version of Google Chrome[23]:

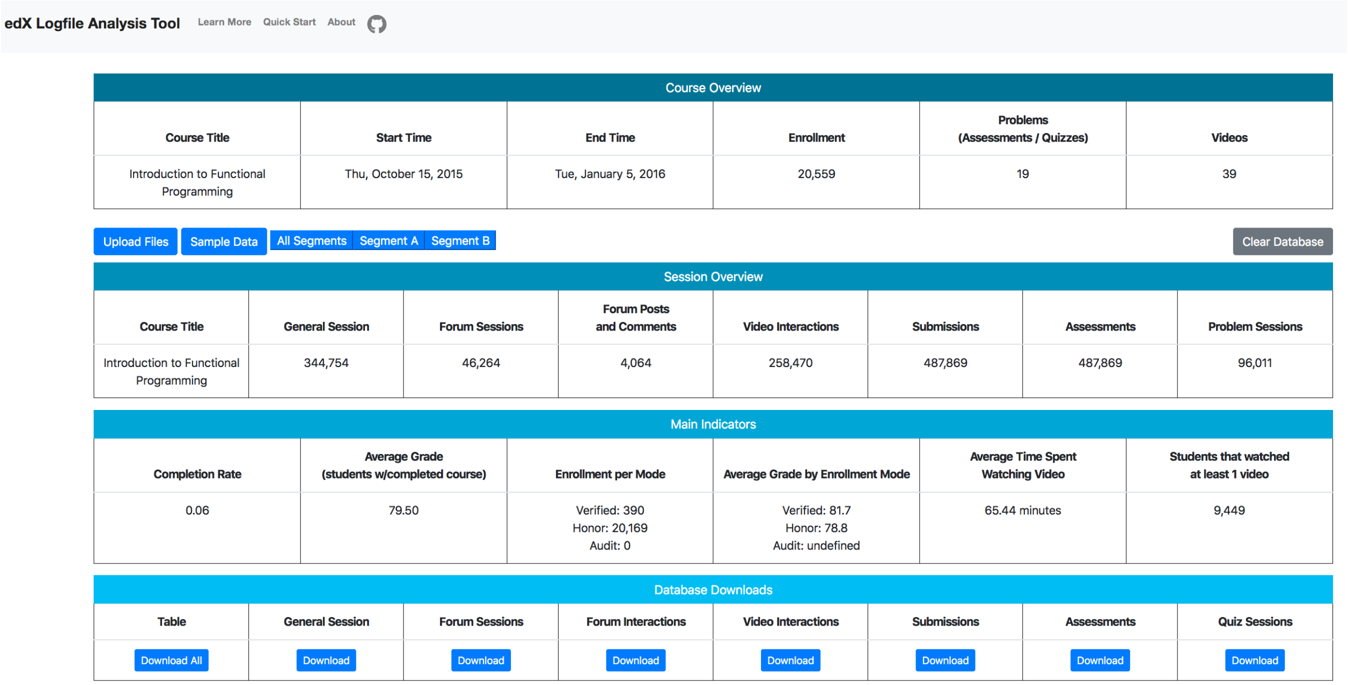
Machine A Windows 10, 16GB RAM, Intel i7 @ 2.80 GHz
Machine B Windows 10, 8GB RAM, Intel i5 @ 2.90 GHz

and found the processing time and disk space use very similar. For each of the MOOCs we report the processing times (in minutes) and occupied disk space (in Gigabytes) in Table 2.

As expected, the disk space usage depends strongly on enrollment and course duration, with the largest MOOC (both in terms of days open and enrollment) requiring about 5 Gigabytes of disk space. On the other hand, the processing time is not only dependent on the duration of the course and the enrollment numbers but also on the *other MOOCs of the same institution* that are running at the same time (last column in Table 2) due to the way the edX log files aggregate all clickstream data of all running MOOCs in one file. Concretely, our smallest MOOC in terms of enrollment (*Robots in society*) has by far the most concurrently running MOOCs (184 over the course of the year) and thus its processing time is almost twice as long as those of the two MOOCs with only 40-50 concurrently running MOOCs. ELAT has to uncompress every log file and read each record in it to determine whether it matches the wanted course identifier. More concretely, the processing time of our four MOOCs varied between one and four hours.

4.1.2 Data Issues. In the process of developing and evaluating ELAT, we encountered a number of data issues, two of which we now describe in greater detail: (i) mobile, and, (ii) open response assessment. They can be attributed to the facts that our MOOCdb

[23] ELAT also works well on Firefox. As ELAT makes use of some rather recently introduced web APIs not all browsers support all web APIs completely as of the time of writing.



Course Overview

Course Title	Start Time	End Time	Enrollment	Problems (Assessments / Quizzes)	Videos
Introduction to Functional Programming	Thu, October 15, 2015	Tue, January 5, 2016	20,559	19	39

Session Overview

Course Title	General Session	Forum Sessions	Forum Posts and Comments	Video Interactions	Submissions	Assessments	Problem Sessions
Introduction to Functional Programming	344,754	46,264	4,064	258,470	487,869	487,869	96,011

Main Indicators

Completion Rate	Average Grade (students w/completed course)	Enrollment per Mode	Average Grade by Enrollment Mode	Average Time Spent Watching Video	Students that watched at least 1 video
0.06	79.50	Verified: 390 Honor: 20,169 Audit: 0	Verified: 81.7 Honor: 78.8 Audit: undefined	65.44 minutes	9,449

Database Downloads

Table	General Session	Forum Sessions	Forum Interactions	Video Interactions	Submissions	Assessments	Quiz Sessions
Download All	Download	Download	Download	Download	Download	Download	Download

Figure 4: ELAT's user interface: data can be uploaded to the browser for processing. Sample data is available as well. Once data has been processed basic course statistics are presented. Data can be filtered according to different conditions (e.g., based on learner IDs to evaluate randomized controlled trials). Data can be filtered and downloaded according to session type.

Table 2: Overview of the courses employed in our technical evaluation. The final column lists the number of MOOCs that ran concurrently to the MOOC being evaluated.

Course	Time Period	#Days	#Enrolled	#Certified Students	#Sessions	Processing Time (in minutes)	Disk Space (in GB)	#MOOCs
Robots in society	04/2018-03/2019	322	2,668	18	20,371	129	0.06	184
Creating powerful political messages	01/2016-01/2017	353	15,002	238	114,500	77	0.41	49
Functional programming	10/2015-01/2016	83	20,559	1149	344,754	54	1.23	40
Solar energy	08/2016-08/2018	731	64,667	582	845,263	324	5.08	78

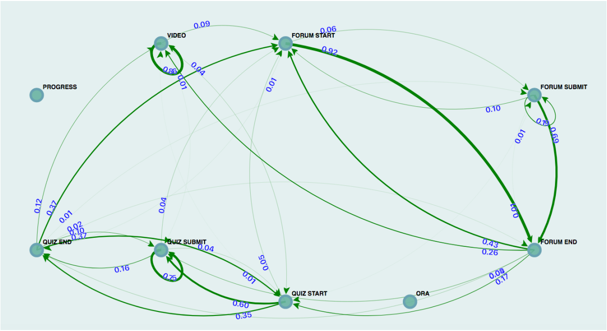


Figure 5: ELAT's learning path visualization for the *Functional Programming* course, based on [2].

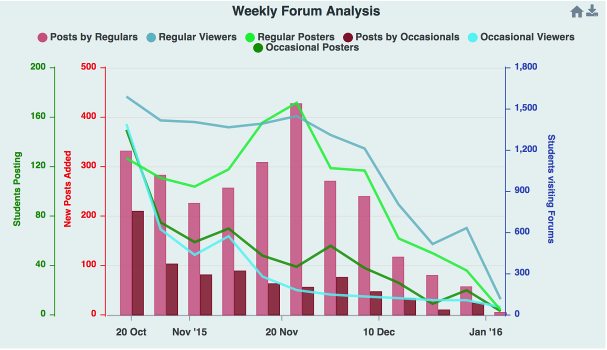


Figure 6: ELAT's forum analysis for the *Functional Programming* course, based on [25]

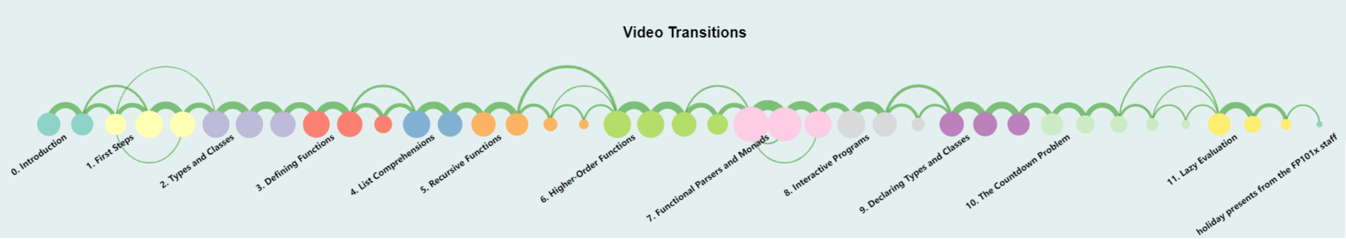


Figure 7: ELAT's video watching sequence visualization for the *Functional Programming* course, based on [2].

code base was developed a number of years ago and the lack of documentation on edX's parts.

The sharing of log data is not one of edX's priorities, as reflected by the small amount of provided documentation. For instance, the changes in the logs' structure is shared with researchers in the edX Release Notes[24], but the last update (at the time of writing this paper) is March 2017. Consequently, special cases in the records have to be handled as they appear.

Concretely, edX mobile (a native app for Android/iOS) was introduced in 2015. Learners using the mobile app can be distinguished by the slightly different logs they generate (one may expect a particular mobile flag in the logs, but this is not the case). Specifically (as we found by digging into the log file format), the format of the `details` field of a video record is a string for a non-mobile interaction, while for a mobile record it is a nested record (JSON)—and thus, ELAT has to identify the type of video record and process the detail accordingly. This is just one example of the edX log file format (slightly) changing over time.

Our second issue refers to the open response assessment (ORA). This type of assessment exists in edX but was not part of the MOOCdb code base. As in our initial interview with potential end users the importance of ORA data came up repeatedly, we extended the MOOCdb data model to incorporate ORA events as well. In this manner, ELAT also aggregates information on the number of times the learner saved his/her solution during the session, if they submitted a solution during the session, how many peers they reviewed, etc.

4.2 User Evaluation

For our user evaluation, we recruited seven participants through internal mailing lists and academic contacts that were required to have some experience in the area of learning analytics, preferably having worked with edX data logs before. The participants completed the study online—we provided each participant with a link to ELAT they could open in their browser as well as a link to the online form we used for the pre/post questionnaire. Each participant was asked to complete three setps:

(1) fill in an initial questionnaire:basic demographics, background and experience;
(2) execute a task: the ELAT version we provided included edX log files of the *Functional Programming* MOOC (cf. Table 2) and participants were asked to explore the data with ELAT and write down 10 insights they gained;
(3) complete an exit questionnaire: we asked participants about the major insights they gained about the MOOC, their opinions about specific features of ELAT and employed the user experience questionnaire [15].

We did not pay our participants and advised them that the experiment would take about an hour of their time. Our participants spend on average 33 minutes in step (2).

4.2.1 *Participants.*

All seven participants (four females and 3 males ranging in age between 25 and 44 years) assume a major role in research at their respective universities. Four of whom are data analysts and consider themselves as advanced in analyzing MOOC learner data whereas the other three consider themselves as novices in this respect. Except for two participants, all have experience in working with edX data.

Our participants work with MOOC learner data for various reasons: ranging from designing feedback interventions to investigating learners' engagement, learning behavior and learning outcomes. Participants cited the understanding of the log data format, cleaning and preparing data for analysis, identifying relevant insights in the sea of data and visualizing the results as major challenges in the analysis of learner data.

4.2.2 *User Experience Questionnaire (UEQ).*

For a standard measure of the participants' experience with ELAT, we applied the well-known User Experience Questionnaire [15]. It consists of 26 bipolar items rated on a 7-point Likert scale, to measure 6 factors of user experience: *attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty.* The results obtained from the questionnaire are shown in Figure 8. Considering the small number of participants, this is not a conclusive result, however the UEQ has been shown to be applicable to small groups [31], and it is useful to form an idea of ELAT's strengths and weaknesses. ELAT scores particularly high on the *stimulation* and *novelty* factors; this is very motivating for the future development of ELAT as we envision ELAT not to be the final step in a learning research effort but instead as a means to conduct this research more efficiently, develop research hypotheses and explore the data. The least positive results on *perspicuity* (i.e., clarity of presentation) can be explained by the relatively small amount of time (33 minutes on average) participants spend on the tool in comparison to the amount of information (both in tabular as well as plotted form) available to them, as well as the fact that ELAT is a prototype and in need of a few more iterations to improve the visualizations' intuitiveness. The high error margins (portrayed by the black lines in Figure 8) can be explained by the difference in

[24] https://edx.readthedocs.io/projects/edx-release-notes/en/latest/

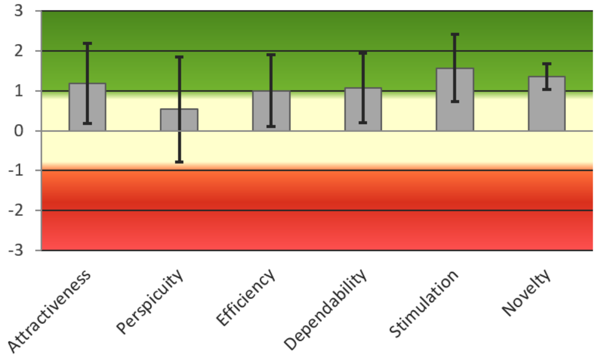


Figure 8: Average UEQ Results for ELAT

self-reported learning data analysis experience: four participants consider themselves advanced while three self-identify as novices.

4.2.3 Insights gained with ELAT*.* In the exit questionnaire we asked our participants to list their ten most important insights obtained about the MOOC. The three recurring themes were forum interaction, video transition and learning paths:

Forum interaction Participants found the forum participation analytics (regularity of viewing, posting and size of posts) and the graphic visualizations very useful to see correlations between regular viewers/posters, occasional viewers/posters and their final grades. Further, participants appreciated the customization of the visual analytics according to desired time frame e.g., weekly, monthly which afforded further investigation on learners' allocation of time in viewing and posting activities and its implications on their final grades.

Video transition The provision of video transition analytics for passing and failing learners enabled participants to gain in-depth information on how and when the two groups of learners adhere/deviate from the prescribed path in video watching. Participants were able to state some core results, e.g., one participant wrote that *"learners who watched the videos sequentially have a higher chance of passing"* and another found the video interaction graph very useful as *"it shows the difference between the instructor views on the sequence of videos and the reaction/perception of learners to this sequence in practice."*

Learning paths Participants found the learning paths analytics useful for comparing how (and when) passing and failing learners interact with the course elements such as video, forum and quiz. For instance, one participant commented that the transition between forum and quiz is higher for failing learners which might imply that they are seeking help in the forum. Another participant found the learning path feature useful in investigating how and when failing learners deviate from the designed learning path which could enable possible interventions or remediations before learners dropout or fail the course.

4.2.4 Features and Functionalities. A number of questions in the exit questionnaire pertained to the features and functionality of ELAT as an analysis tool. Participants' comments chiefly revolved around the customizable and downloadable graphs, downloadable session database and the main indicators table (as seen in Figure 4). Most of the participants were positive except for one who found the features confusing, in particular, the graphs. Five participants found the overview and main indicators table helpful for quick access to descriptive statistics and course information. Two participants commented that the downloadable session database facilitates the transformation of data logs into comma-separated values (csv) files which afford further in-depth analysis and another found it complemented the current instrument he used in the analysis of MOOC data. Five participants found the customizable and downloadable graphs for forum interaction, video transition and learning paths were most useful and effective amongst all the available features of ELAT.

4.2.5 Recommendations. Most of the participants' suggestions for the improvement of ELAT centered on the visual analytics of the forum participation, video interaction and learning path. Participants requested for provision of more in-depth information e.g., the distribution of failing learners in their engagement with the various course components. The selection of specific segment of learners' engagement e.g., in video interaction could provide more insightful information on when failing learners display disengagement with course content. One participant recommended the inclusion of content analysis and social network analysis features to afford a more wholesome investigation of the patterns and trends in forum activities. Other suggestions involved format aspects such as labeling, legends and colour schemes of the line graphs and tables, as well as provision of course structure elements in the place of timeline in the x-axis. The latter will afford an evaluation of learners' behavioral responses in relation to the course elements. Likewise, the type of enrollment (audit, honor or verified) apart from passing and failing could better inform learners' behavior in their interactions with the various course elements. Overall, most of the participants see the potential in ELAT and express keen interest in using ELAT for future analysis of MOOC learner data.

5 CONCLUSIONS

This paper presents ELAT, an open-source, privacy-aware and browser-based edX log data analysis tool. The primary goal in the design and development of ELAT is to equip and to empower researchers with less data processing experience to effectively and efficiently analyse large MOOC data sets—in contrast to existing tools, ELAT requires no computational knowledge to set it up.

Our system evaluation has shown that edX MOOCs with tens of thousands of learners can be analyzed within the browser, within a few hours of time on a standard machine.

From our user study, it is evident that the value-added features of ELAT (forum interaction, video transition and learning paths) have enabled researchers to tease out important findings on learners' learning progress as well as learning experience easily and effectively.

Most of our participants found the customizable visualizations of learner engagement in course activities and the downloadable session data for further analysis particularly useful. On the same note, these features would also be useful for instructional designers to make appropriate design decisions for various target learners and learning objectives. Likewise, for course facilitators, ELAT could be instrumental to monitor learning progress and to appropriate effective measures of interventions when learners show signs of prolonged disengagement.

In the future we plan to take up our participants' suggestions and conduct a more thorough user study. Specifically, we are planning a replication study by asking learning researchers to reproduce (part of) their previous work on edX log data with ELAT, allowing us to assess their performance and experience in a more natural study setup.

REFERENCES

[1] Y. Chen, Q. Chen, Mingqian Zhao, S. Boyer, K. Veeramachaneni, and H. Qu. 2016. DropoutSeer: Visualizing learning patterns in Massive Open Online Courses for dropout reasoning and prediction. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST).* 111–120.

[2] Dan Davis, Guanliang Chen, Claudia Hauff, and Geert-Jan Houben. 2016. Gauging MOOC Learners' Adherence to the Designed Learning Path. *International Educational Data Mining Society* (2016).

[3] Dan Davis, Guanliang Chen, Claudia Hauff, and Geert-Jan Houben. 2018. Activating learning at scale: A review of innovations in online learning strategies. *Computers & Education* 125 (2018), 327–344.

[4] Dan Davis, Guanliang Chen, Ioana Jivet, Claudia Hauff, and Geert-Jan Houben. 2016. Encouraging Metacognition & Self-Regulation in MOOCs through Increased Learner Feedback.. In *LAL@ LAK.* 17–22.

[5] Dan Davis, Vasileios Triglianos, Claudia Hauff, and Geert-Jan Houben. 2018. Srlx: A personalized learner interface for moocs. In *European Conference on Technology Enhanced Learning.* Springer, 122–135.

[6] Sara Isabella De Freitas, John Morgan, and David Gibson. 2015. Will MOOCs transform learning and teaching in higher education? Engagement and course retention in online learning provision. *British Journal of Educational Technology* 46, 3 (2015), 455–471.

[7] Oliver Ferschke, Diyi Yang, Gaurav Tomar, and Carolyn Penstein Rosé. 2015. Positive impact of collaborative chat participation in an edX MOOC. In *International Conference on Artificial Intelligence in Education.* Springer, 115–124.

[8] S. Fu, J. Zhao, W. Cui, and H. Qu. 2017. Visual Analysis of MOOC Forums with iForum. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 201–210.

[9] Jeffrey A Greene, Christopher A Oswald, and Jeffrey Pomerantz. 2015. Predictors of retention and achievement in a massive open online course. *American Educational Research Journal* 52, 5 (2015), 925–955.

[10] Philip J Guo, Juho Kim, and Rob Rubin. 2014. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the first ACM conference on Learning@ scale conference.* ACM, 41–50.

[11] Kate S Hone and Ghada R El Said. 2016. Exploring the factors affecting MOOC retention: A survey study. *Computers & Education* 98 (2016), 157–168.

[12] Juho Kim, Philip J Guo, Daniel T Seaton, Piotr Mitros, Krzysztof Z Gajos, and Robert C Miller. 2014. Understanding in-video dropouts and interaction peaks inonline lecture videos. In *Proceedings of the first ACM conference on Learning@ scale conference.* ACM, 31–40.

[13] René F Kizilcec, Mar Pérez-Sanagustín, and Jorge J Maldonado. 2016. Recommending self-regulated learning strategies does not improve performance in a MOOC. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale.* ACM, 101–104.

[14] René F Kizilcec, Chris Piech, and Emily Schneider. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge.* ACM, 170–179.

[15] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work*, Andreas Holzinger (Ed.). Berlin, Heidelberg, 63–76.

[16] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. 2015. MOOC video interaction patterns: What do they tell us? In *Design for teaching and learning in a networked world.* 197–210.

[17] Allison Littlejohn, Nina Hood, Colin Milligan, and Paige Mustain. 2016. Learning in MOOCs: Motivations and self-regulated learning in MOOCs. *The Internet and Higher Education* 29 (2016), 40–48.

[18] Glenn Lopez, Daniel T Seaton, Andrew Ang, Dustin Tingley, and Isaac Chuang. 2017. Google BigQuery for education: framework for parsing and analyzing edX MOOC data. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale.* ACM, 181–184.

[19] Colin Milligan, Allison Littlejohn, and Anoush Margaryan. 2013. Patterns of engagement in connectivist MOOCs. *Journal of Online Learning and Teaching* 9, 2 (2013), 149–159.

[20] Zachary A. Pardos and Kevin Kao. 2015. moocRP: An Open-source Analytics Platform. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale (L@S '15).* New York, NY, USA, 103–110.

[21] Zachary A Pardos, Steven Tang, Daniel Davis, and Christopher Vu Le. 2017. Enabling real-time adaptivity in MOOCs with a personalized next-step recommendation framework. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale.* ACM, 23–32.

[22] Zachary A Pardos, Anthony Whyte, and Kevin Kao. 2016. moocRP: enabling open learning analytics with an open source platform for data distribution, analysis, and visualization. *Technology, Knowledge and Learning* 21, 1 (2016), 75–98.

[23] Trang Phan, Sara G McNeil, and Bernard R Robin. 2016. StudentsâĂŹ patterns of engagement and course performance in a Massive Open Online Course. *Computers & Education* 95 (2016), 36–44.

[24] Héctor J Pijeira Díaz, Javier Santofimia Ruiz, José A Ruipérez-Valiente, Pedro J Muñoz-Merino, and Carlos Delgado Kloos. 2016. A demonstration of ANALYSE: a learning analytics tool for open edX. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale.* ACM, 329–330.

[25] Oleksandra Poquet, Vitomir Kovanović, Pieter de Vries, Thieme Hennis, Srećko Joksimović, Dragan Gašević, and Shane Dawson. 2018. Social presence in massive open online courses. *International Review of Research in Open and Distributed Learning* 19, 3 (2018).

[26] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. 2013. Modeling learner engagement in MOOCs using probabilistic soft logic. In *NIPS Workshop on Data Driven Education*, Vol. 21. 62.

[27] Justin Reich and José A. Ruipérez-Valiente. 2019. The MOOC pivot. *Science* 363, 6423 (2019), 130–131.

[28] Jan Renz, Daniel Hoffmann, Thomas Staubitz, and Christoph Meinel. 2016. Using A/B testing in MOOC environments. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge.* ACM, 304–313.

[29] Tarmo Robal, Yue Zhao, Christoph Lofi, and Claudia Hauff. 2018. IntelliEye: Enhancing MOOC Learners' Video Watching Experience through Real-Time Attention Tracking. In *Proceedings of the 29th on Hypertext and Social Media.* ACM, 106–114.

[30] Javier Santofimia Ruiz, Héctor J Pijeira Díaz, José A Ruipérez-Valiente, Pedro J Muñoz-Merino, and Carlos Delgado Kloos. 2014. Towards the development of a learning analytics extension in open edX. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality.* ACM, 299–306.

[31] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. 2014. Applying the user experience questionnaire (UEQ) in different evaluation scenarios. In *International Conference of Design, User Experience, and Usability.* Springer, 383–392.

[32] Daniel T Seaton, Yoav Bergner, Isaac Chuang, Piotr Mitros, and David E Pritchard. 2014. Who does what in a massive open online course? (2014).

[33] Conglei Shi, Siwei Fu, Qing Chen, and Huamin Qu. 2015. VisMOOC: Visualizing video clickstream data from massive open online courses. In *2015 IEEE Pacific visualization symposium (PacificVis).* IEEE, 159–166.

[34] Kalyan Veeramachaneni, Franck Dernoncourt, Colin Taylor, Zachary Pardos, and Una-May OâĂŹReilly. 2013. Moocdb: Developing data standards for mooc data science. In *AIED 2013 workshops proceedings volume*, Vol. 17. Citeseer.

[35] Kalyan Veeramachaneni, Sherif Halawa, Franck Dernoncourt, Una-May O'Reilly, Colin Taylor, and Chuong Do. 2014. Moocdb: Developing standards and systems to support mooc data science. *arXiv preprint arXiv:1406.2015* (2014).

[36] Sunnie Lee Watson, William R Watson, Ji Hyun Yu, Hamdan Alamri, and Chad Mueller. 2017. Learner profiles of attitudinal learning in a MOOC: An explanatory sequential mixed methods study. *Computers & Education* 114 (2017), 274–285.

[37] Miaomiao Wen, Diyi Yang, and Carolyn Rose. 2014. Sentiment Analysis in MOOC Discussion Forums: What does it tell us?. In *Educational data mining 2014.* Citeseer.

[38] Julie Wintrup, Kelly Wakefield, and Hugh C Davis. 2015. Engaged learning in MOOCs: a study using the UK Engagement Survey. (2015).