Academic Time Tracker

Miguel Rochefort
miguel.rochefort@gatech.edu

***Abstract*—**Time on task is defined as the total amount of time a student spends on an academic task or course. It is considered to be one of the best indicators of learning. Yet, little is known about how students actually spend their time. Indeed, time-on-task estimation from learning management system trace data can only capture a small fraction of a student's learning activities. While some client-side tracking solutions claim to achieve better coverage, they typically are not validated against ground truth. In this paper, we propose the Academic Time Tracker, a tool that automatically detects and classifies learning sessions from computer usage logs. When tested against the logs of a student completing 31 online courses across 500+ ground-truth learning sessions, our method achieved a coverage of 96%, a detection accuracy of 85%, and a weighted classification accuracy of 80%, supporting client-side tracking as a promising alternative to server-side tracking for time-on-task estimation.

1 INTRODUCTION

In education research, time-on-task is defined as the amount of time a student spends on a given learning task. It is thought to be one of the best indicators of learning (Stallings, 1980). Over the years, researchers have used time-on-task data to predict course completion and withdrawal (Morris et al., 2005), identify patterns of struggling students (Nguyen et al., 2018), evaluate the quality of learning sessions (Vicente & Hellas, 2021), and help instructors adjust their course workload (Holstein et al., 2018).

Unfortunately, accurate time-on-task data is difficult to obtain. In 1982, Karweit & Slavin identified many of the challenges associated with gathering time-on-task data. As a result, many studies rely on self-reported time-on-task estimates, which are widely known to be inaccurate (Parry et al., 2021; Junco, 2013; Rosen et al., 2017).

In recent years, the learning analytics community has moved away from these inaccurate self-reports towards supposedly more objective server-side trace data. However, it is not clear whether this shift represents an actual improvement in time-on-task accuracy. Indeed, researchers have raised concerns about the inconsistency of time-on-task estimation methods using trace data (Kovanovic et al., 2015). Despite attempts to compare different methods (Papamitsiou et al., 2016) and improve their accuracy (Halvoník et al., 2021; Nguyen, 2020; Rushkin, 2018), these systems still fail to capture the majority of learning activities because they take place outside learning management systems (LMS) where trace data is not collected (García et al., 2012; Verbert et al., 2012).

These shortcomings call for novel time-on-task estimation methods that provide better coverage and accuracy. In this paper, we propose the Academic Time Tracker, a tool that automatically detects and classifies learning sessions from client-side computer usage logs. We evaluate our solution against our own Computer Science Student Activity Dataset and find it to be a promising alternative to LMS trace data for time-on-task estimation.

2.1 Experience API

Students generally use more than one LMS, making it challenging to get a complete picture of their learning habits. To address this issue, the United States Department of Defense sponsored the development of the Experience API (xAPI), allowing data from multiple LMS to be combined into a single Learning Record Store (LRS) ("Experience API," 2020). Kitto et al. (2015) developed the Connected Learning Analytics (CLA) toolkit, which combined learning activities from multiple social media platforms into a single LRS. Sheshadri et al. (2019) aggregated traces from forums, learning management systems, homework systems, and version control systems to better understand student behavior.

2.2 Proxy

Jaakonmäki et al. (2020) introduced the Web Analytics Proxy Service, which adds tracking capabilities to arbitrary websites by replacing URLs and inserting activity tracking scripts. While this solution requires no setup for the user, it is limited to web pages visited through a dedicated portal and does not support tracking for dynamic or authenticated websites.

2.3 Browser history

Kovacs (2021) proposed a machine learning algorithm that reconstructs detailed browsing activity from readily available browser history with a 76.2% accuracy. While it can leverage months of browser interaction without any prior setup, it is limited to learning activities taking place inside the browser.

2.4 Virtual machine

Pardo & Kloos (2011) and García et al. (2012) used virtual machines to track all learning activities taking place within them. While accurate, their complex setup makes them impractical for students using their own computers.

2.5 Screen recording

Bortoluzzi & Marenzi (2017) used screen recording to analyze the search engine habits of language instructors but used humans for annotation. Krieter (2020) & Breiter (2018) combined mobile screen recording with computer vision and machine learning to automatically detect learning activities. However, their method is manual, consumes significant power and storage, and is limited by the state of computer vision and machine learning.

2.6 Activity tracking

Santos et al. (2012) used the RescueTime activity tracking software to study the learning patterns of engineering students. Kimmons et al. (2017) developed a browser extension to track and analyze the browsing habits of teachers, teacher candidates, and K-12 students. Unfortunately, none investigated the detection and classification of learning activities.

2.7 Multimodal

Di Mitri et al. (2017) proposed Learning Pulse, a multimodal system combining heart rate and step count (Fitbit), weather condition (OpenWeatherMap), and learning activity (RescueTime) to predict learning performance in self-regulated learning settings. Zhu et al. (2019) introduced LifeLogger, a web application combining fitness (Fitbit), location (Moves), and computer activity (RescueTime) data to help college students reflect on their use of time. Kong et al. (2020) presented a semi-automated Day Reconstruction Method using 24-hour data collected from a wearable (Fitbit Versa), mobile phone (Red Mi 6), and activity tracking software (RescueTime) to measure academic engagement.

3 RESEARCH QUESTIONS

To properly understand the viability and limits of client-side academic tracking and evaluate the performance of the Academic Time Tracker, we identified three research questions that we consider essential to answer.

3.1 RQ1 Coverage: What fraction of learning sessions produce client-side trace data?

The purpose of this question is to understand the practical limits of client-side tracking. For the Academic Time Tracker to be effective, trace data must exist during most learning sessions. If we find that an important fraction of learning activities take place offline where no trace data is captured (e.g., physical books), this would significantly limit the potential of our approach.

3.2 RQ2 Detection: How accurately can learning sessions be detected using client-side trace data?

The purpose of this question is to understand whether trace data contain enough information for the Academic Time Tracker to detect learning activities accurately. We hypothesize that specific applications (e.g., Microsoft Word) and domains (e.g., https://study.com) are strong predictors of academic learning and could be used to detect learning sessions.

3.3 RQ3 Classification: How accurately can learning sessions be classified into courses using client-side trace data?

The purpose of this question is to understand whether trace data contains enough information for the Academic Time Tracker to classify the course associated with a learning activity accurately. We hypothesize that URLs and titles contain information (e.g., C958) and keywords (e.g., calculus) that could be used to classify learning sessions accurately.

4 METHOD

4.1 Dataset

From June 2020 to November 2020, an undergraduate computer science student completed 31 online courses across three institutions. While studying, they manually recorded every learning session in Google Calendar and automatically recorded all computer activities using ActivityWatch. This data was combined to create the Computer Science Student Activity Dataset (CSSAD).

Table 1—Overview of the Computer Science Student Activity Dataset (CSSAD).

Dataset	Columns	Rows
Course details	title, code, provider, syllabus, urls, start, end	31
Google Calendar sessions	title, start, end	503
ActivityWatch trace events	title, app, url, start, end	520,920

4.1.1 Courses

To complete their Bachelor of Science in Computer Science degree, the student had to complete 31 online courses across three US-accredited online education institutions: Study.com, Sophia Learning, and Western Governors University. The list of all the completed courses can be found in Appendix 9.1: Courses. The details of these courses, including their title, code, provider, syllabi (PDF), start date, and end date, are included in the CSSAD dataset.

4.1.2 Google Calendar

While studying, the student manually recorded every learning session in Google Calendar. A total of 503 events were recorded. Each event includes a start time, an end time, and a title containing the course's code. The student rounded the start and end times to the nearest 15-minute mark for convenience. The data was exported using the Google Calendar API, converted to a CSV file, and added to the CSSAD dataset. This will serve as the ground truth for evaluating the performance of the Academic Time Tracker.

4.1.3 ActivityWatch

While studying, the student automatically recorded all their computer activity using the open-source ActivityWatch activity tracker. The software collected trace data from their Lenovo ThinkPad T420 (Windows), Teclast F5 (Windows), Nokia 6.1 (Android), and Google Chromecast (TV). A total of 520,920 events were recorded. Each event includes a start time, an end time, an application name, a title, and an optional URL. The trace data was exported from the ActivityWatch dashboard of each device, combined into a single CSV file, and added to the CSSAD dataset.

4.1.4 Labels

To facilitate the training of a machine learning classifier able to identify the course associated with a given trace event, we manually labeled a total of 91,424 trace events with one of 40 academic classes. These target labels were added to the CSSAD dataset. See Appendix 9.2: Labels for a list of all label classes and associated example trace events.

4.2 Solution

The Academic Time Tracker automatically detects and classifies learning sessions from computer usage logs in two steps: (1) classify individual trace events according to their predicted class using a machine learning model and (2) cluster individual trace events into class-coherent learning sessions of 15-minute increments using a sliding window algorithm.

4.2.1 Classification

While labeling ActivityWatch trace events, we noticed that only three features were useful in determining their class: app, URL, and title. To tokenize and embed these features, we used scikit-learn's `CountVectorizer` and `TfidfTransformer`. For classification, we opted for scikit-learn's `LinearSVC` because it provided the best performance without requiring parameter tuning. After training and testing our classifier (70-30 shuffled split), we achieved a weighted accuracy, precision, recall, and F1 score of 99%. We used the classifier to classify all individual trace events into one of 40 classes.

4.2.2 Clustering

After classifying all trace events, we need to cluster them into coherent sessions that would be recognizable to a student. For example, a 1-minute calculus session followed by a 1-minute break followed by another 1-minute calculus session should be avoided. Because our ground truth Google Calendar learning sessions are rounded and aligned to the nearest 15-minute mark, we designed a sliding window clustering algorithm using a 15-minute window size.

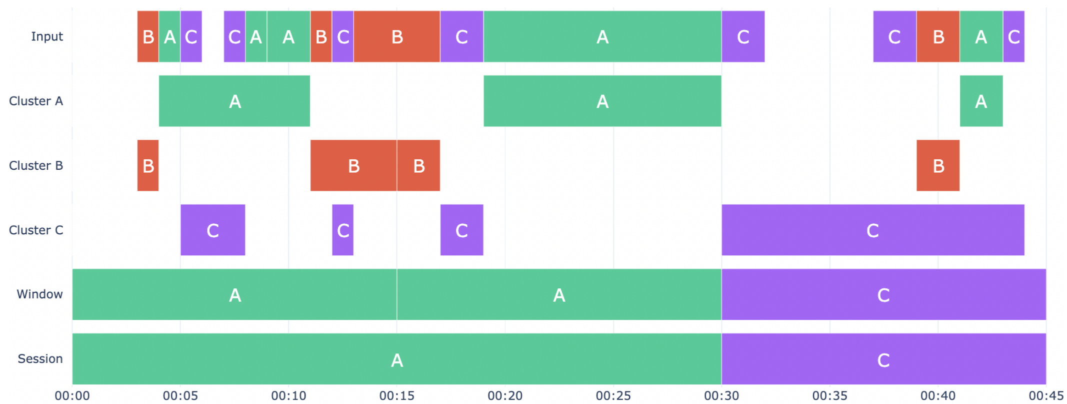

Figure 1—Sliding window clustering algorithm.

Within each 15-minute window, we clustered trace events of the same class separated by at most 5 minutes. Then, we calculated the window coverage of each class by taking the sum of the duration of their clusters. We considered the user inactive if the maximum class coverage was less than 50% of the window size (7.5 minutes). Otherwise, we multiplied each class coverage by a weight representing their specificity. For example, we assigned class *c958* a larger weight than *study.com* because it is more specific. Then, we assigned the class with the maximum weighted coverage to each 15-minute window. Finally, consecutive 15-minute windows with the same class were merged into final predicted sessions of 15-minute increments.

4.3 Evaluation

To evaluate the performance of the Academic Time Tracker, we compared the predicted sessions to the actual sessions from Google Calendar. First, we filled the gaps between sessions with sessions of class *afk*. Then, we segmented and compared predicted and actual sessions using different rules to reflect each research question (illustrated in Figure 2).

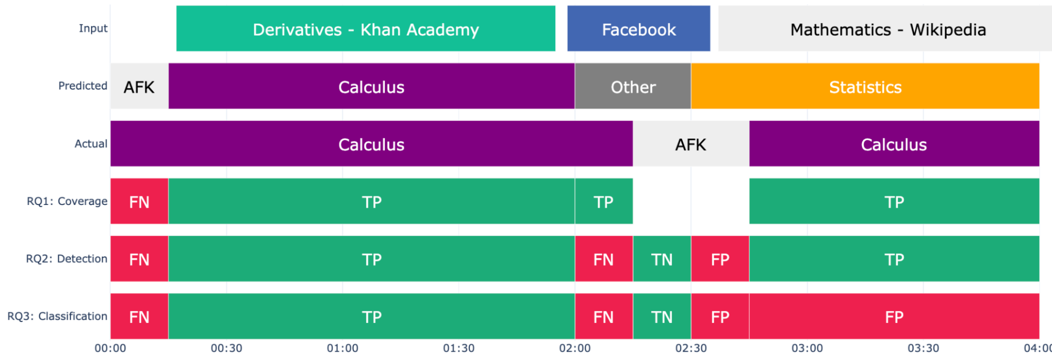

Figure 2—Comparing predicted sessions to actual sessions.

RQ1 Coverage—Class *afk* is considered negative (N). All other classes are considered positive (P) and equal to each other (T) because any session is a potential study session. True negatives (TN) and false positives (FP) are ignored because misdetection and over-coverage are not problems.

RQ2 Detection—Classes *afk* and *other* are considered negative (N) and equal to each other (T). All other classes are considered positive (P) and equal to each other (T) because we are only concerned with detecting learning sessions, not classifying them.

RQ3 Classification—Classes *afk* and *other* are considered negative (N) and equal to each other (T). All other classes are considered positive (P) but not equal to each other (F) because we are concerned with classifying learning sessions, not just detecting them.

Finally, we used scikit-learn's `classification_report` to measure the accuracy, precision, recall, and F1 score associated with each research question.

5 RESULTS

Evaluating predicted sessions against actual sessions according to the methods described in Section 4.3 produced the results presented in Table 2.

Table 2—Performance of

Research Question	Accuracy	Precision	Recall	F1 Score
RQ1: Coverage	0.9632	-	-	-
RQ2: Detection	0.8478	0.8160	0.8535	0.8286
RQ3: Classification	0.7996	0.5656	0.5249	0.5178

5.1 RQ1 Coverage: What fraction of learning sessions produce client-side trace data?

When comparing predicted sessions from all trace events to actual learning sessions, over 96% coverage was measured. In other words, 580 out of 602 hours of learning sessions produced trace data. Trace events from the desktop, browser, smartphone, and smart TV achieved a coverage of 76.20%, 72.05%, 37.30%, and 12.37%, respectively. Traces containing URLs associated with LMS (see Appendix 9.3: Learning Management System domains) covered 31.08% of learning sessions. A breakdown of the student's most used learning resources can be found in Appendix 9.4. Figure 3 illustrates the coverage provided by different trace data sources.

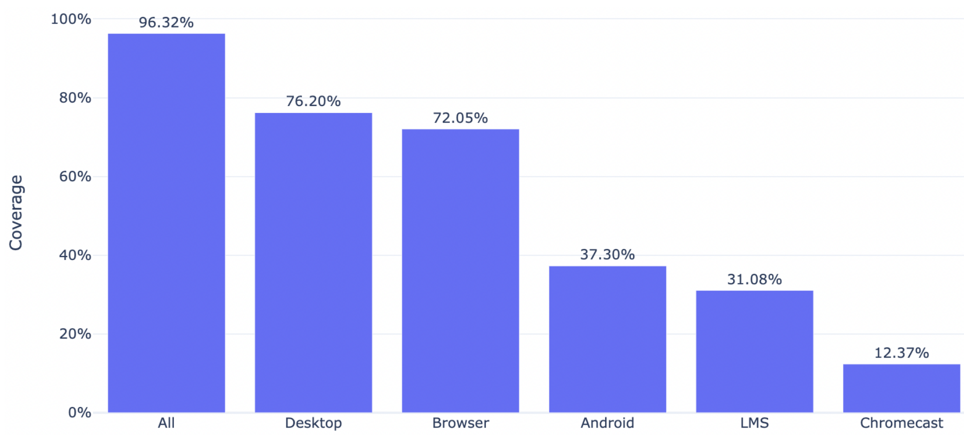

Figure 3—Coverage of learning sessions using different sources.

5.2 RQ2 Detection: How accurately can learning sessions be detected using client-side trace data?

When comparing predicted sessions to actual learning sessions, our method achieved a detection accuracy of 85%, a precision of 82%, a recall of 85%, and an F1 score of 83%. While our solution accurately detected 1205 hours of non-learning sessions and 523 hours of learning sessions, it mistakenly found an extra 230 hours and missed another 80 hours of learning, as seen in Table 3.

Table 3—Confusion matrix of learning session detection.

	Predicted Negative	Predicted Positive
Actually Negative	1205 hours	230 hours
Actually Positive	80 hours	523 hours

5.3 RQ3 Classification: Classification: How accurately can learning sessions be classified into courses using client-side trace data?

When comparing predicted sessions to actual sessions, our method achieved a classification accuracy of 80%, a precision of 57%, a recall of 52%, and an F1 score of 52%. Figure 4 compares predicted and actual time spent on each class.

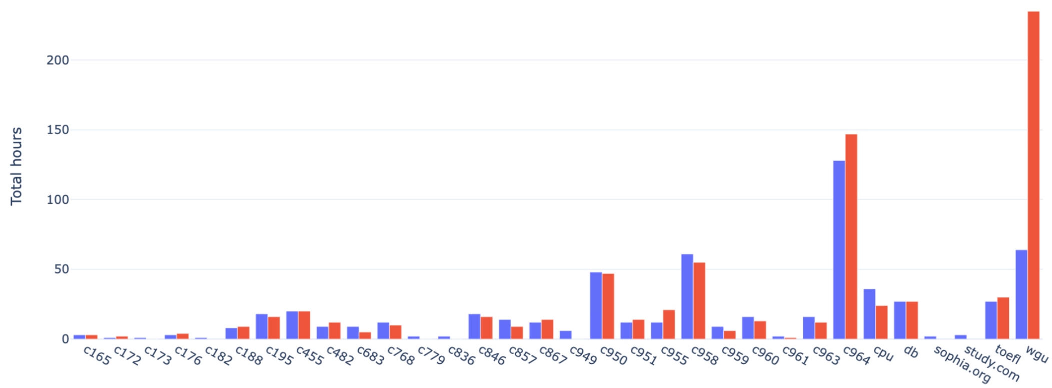

Figure 4—Predicted (blue) vs. Actual (red) time spent on a class.

6 DISCUSSION

In this paper, we proposed the Academic Time Tracker, a tool that automatically detects and classifies learning sessions from computer usage logs. The tool's goal was to provide better coverage than LMS while maintaining accuracy for time-on-task estimation. The tool was designed to use data from existing activity tracking software such as ActivityWatch, which we predicted would capture most learning activities. This discussion will cover the results of our research questions, the limitations of our approach, and our planned future work.

6.1 Research question results

6.1.1 RQ1 Coverage: What fraction of learning sessions produce client-side trace data?

To address RQ1, we looked for the presence of trace data during learning sessions. We found that trace data covered 96% of study sessions, which exceeded our expectations. We were also surprised to find that Android trace data alone captured 37% of study sessions, which combined with 76% desktop coverage, indicates that many sessions involved multiple devices and supports the growing interest for mobile learning (Krull & Duart, 2017). Another notable discovery was that trace data with URLs associated with any LMS only achieved a 31% coverage, which confirms that LMS trace data is not a viable approach for total time-on-task estimation. These results alone make a strong case for the use of client-side trace data for time-on-task estimation.

6.1.2 RQ2 Detection: How accurately can learning sessions be detected using client-side trace data?

To address RQ2, we examined the Academic Time Tracker's ability to differentiate learning sessions from non-learning sessions. We measured an accuracy of 85%, which appeared quite impressive at first. However, the confusion matrix (Table 3) clearly shows that 230 hours of non-learning sessions were misclassified as learning sessions, which is a non-trivial error size not immediately reflected in the reported accuracy. This is likely caused by an imbalance in learning and non-learning sessions, which skewed the results positively. Adjusting our metrics and algorithm to better reflect and address the high rate of false positives would make a stronger case for the tool's detection performance.

6.1.3 RQ3 Classification: How accurately can learning sessions be classified into courses using client-side trace data?

To address RQ3, we examined the Academic Time Tracker's ability to classify learning sessions according to their associated course. We measured a weighted accuracy of 80%, which is very impressive given the 40 classes possible. A look at the comparison between predicted and actual time spent on each class (Figure 4) shows a very close match for most classes, aside from a few minor classes reporting zero hours and *wgu* being overestimated by a factor of 4. Tweaking the trace event classifier to be less sensitive to the *wgu* class might reduce false positives and improve the accuracy of both RQ2 and RQ3. The fact that only a single class appears to be significantly misclassified is definitely encouraging.

6.2 Limitations

Although our tool achieved promising results for our particular case study, it is difficult to predict how it would translate to other students. First, the student only attended fully online courses, which likely reduced the number of learning activities taking place offline. Second, they studied computer science, a field in which most learning activities require a computer. Third, the student only studied using devices that supported the ActivityWatch tracker, which is not the case for many popular devices such as the iPhone and iPad. Fourth, the student was studying full-time and not working, which limited the amount of trace data that could have confused the classifier given that many tools are used both in an academic and professional setting. Finally, the classifier was only trained to recognize the specific courses, programs, and schools attended by the student. More work and data are needed to support other courses and make the tool usable for other students.

6.3 Future Work

In the future, we plan to improve the accuracy of the tool, with a focus on reducing the number of false positives detected. We believe this could be done by resolving dataset imbalances using techniques such as SMOTE. If a significant improvement is achieved, we will investigate ways to generalize our tool to support unseen courses that the classifier was not trained with. This might require the use of advanced language models such as BERT and topic modeling techniques such as Latent Dirichlet Allocation. We also believe that session segmentation could benefit from hierarchical clustering algorithms such as HDBSCAN. Lastly, we would like to answer the natural successor to our three research questions:

> *RQ4 Identification: How accurately can individual assignments be identified using client-side trace data?*

7 CONCLUSION

Time-on-task estimation from server-side LMS trace data provides an incomplete view of student time, achieving a coverage of just 31% in our case study. In contrast, we found that client-side traces can capture over 96% of total learning time. We proposed the Academic Time Tracker, a tool that used client-side trace data to automatically detect and classify learning sessions, with respective accuracies of 85% and 80%. Our study showed that client-side trace data is a promising alternative to server-side LMS trace data for time-on-task estimation.

8 REFERENCES

1. Bortoluzzi, M., & Marenzi, I. (2017). *Web searches for learning. How language teachers search for online resources* [Data set]. University of Salento. https://doi.org/10.1285/I22390359V23P21
2. Di Mitri, D., Scheffel, M., Drachsler, H., Börner, D., Ternier, S., & Specht, M. (2017). Learning pulse: A machine learning approach for predicting performance in self-regulated learning using multimodal data. *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, 188–197. https://doi.org/10.1145/3027385.3027447
3. Experience API. (2020). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Experience_API&oldid=987066254
4. García, R. M. C., Pardo, A., Kloos, C. D., Niemann, K., Scheffel, M., & Wolpers, M. (2012). Peeking into the black box: Visualising learning activities. *International Journal of Technology Enhanced Learning*, *4*(1/2), 99. https://doi.org/10.1504/IJTEL.2012.048313
5. Halvoník, D., Kapusta, J., & Munk, M. (2021). Improve estimated time-on-task calculation in a Virtual Learning Environment. *Interactive Learning Environments*, *0*(0), 1–16. https://doi.org/10.1080/10494820.2021.1913609
6. Holstein, K., Yu, Z., Sewall, J., Popescu, O., McLaren, B. M., & Aleven, V. (2018). Opening Up an Intelligent Tutoring System Development Environment for Extensible Student Modeling. In C. Penstein Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, & B. du Boulay (Eds.), *Artificial Intelligence in Education* (pp. 169–183). Springer International Publishing. https://doi.org/10.1007/978-3-319-93843-1_13
7. Jaakonmäki, R., vom Brocke, J., Dietze, S., Drachsler, H., Fortenbacher, A., Helbig, R., Kickmeier-Rust, M., Marenzi, I., Suarez, A., & Yun, H. (2020). Understanding Students' Online Behavior While They Search on the Internet: Searching as Learning. In R. Jaakonmäki, J. vom Brocke, S. Dietze, H. Drachsler, A. Fortenbacher, R. Helbig, M. Kickmeier-Rust, I. Marenzi, A. Suarez, & H. Yun (Eds.), *Learning Analytics Cookbook: How to Support Learning Processes Through Data Analytics and Visualization* (pp. 75–88). Springer International Publishing. https://doi.org/10.1007/978-3-030-43377-2_7

8. Junco, R. (2013). Comparing actual and self-reported measures of Facebook use. *Computers in Human Behavior*, *29*(3), 626–631. https://doi.org/10.1016/j.chb.2012.11.007
9. Karweit, N., & Slavin, R. E. (1982). Time-on-task: Issues of timing, sampling, and definition. *Journal of Educational Psychology*, *74*(6), 844–851. https://doi.org/10.1037/0022-0663.74.6.844
10. Kimmons, R., Clark, B., & Lim, M. (2017). Understanding web activity patterns among teachers, students and teacher candidates. *Journal of Computer Assisted Learning*, *33*(6), 588–596. https://doi.org/10.1111/jcal.12202
11. Kitto, K., Cross, S., Waters, Z., & Lupton, M. (2015). Learning analytics beyond the LMS: The connected learning analytics toolkit. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, 11–15. https://doi.org/10.1145/2723576.2723627
12. Kong, R., Hu, X., & Yuen, A. H. K. (2020, January 7). *Understanding Academic Engagement and Context Through Multimodal Data*. https://doi.org/10.24251/HICSS.2020.411
13. Kovacs, G. (2021). Reconstructing Detailed Browsing Activities from Browser History. *ArXiv:2102.03742 [Cs]*. http://arxiv.org/abs/2102.03742
14. Kovanovic, V., Gašević, D., Dawson, S., Joksimovic, S., & Baker, R. (2015). Does Time-on-task Estimation Matter? Implications on Validity of Learning Analytics Findings. *Journal of Learning Analytics*, *2*(3), 81–110. https://doi.org/10.18608/jla.2015.23.6
15. Krieter, P. (2020). *Looking Inside—Mobile Screen Recordings as a Privacy Friendly Long-Term Data Source to Analyze User Behavior*. https://doi.org/10.26092/elib/103
16. Krieter, P., & Breiter, A. (2018). *Track every move of your students: Log files for Learning Analytics from mobile screen recordings*. Gesellschaft für Informatik e.V. http://dl.gi.de/handle/20.500.12116/21042
17. Morris, L. V., Finnegan, C., & Wu, S.-S. (2005). Tracking student behavior, persistence, and achievement in online courses. *The Internet and Higher Education*, *8*(3), 221–231. https://doi.org/10.1016/j.iheduc.2005.06.009
18. Nguyen, Q., Huptych, M., & Rienties, B. (2018). Linking students' timing of engagement to learning design and academic performance. *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, 141–150. https://doi.org/10.1145/3170358.3170398
19. Papamitsiou, Z., Karapistoli, E., & Economides, A. (2016, April 26).

Applying classification techniques on temporal trace data for shaping student behavior models. https://doi.org/10.1145/2883851.2883926
20. Pardo, A., & Kloos, C. D. (2011). Stepping out of the box: Towards analytics outside the learning management system. *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 163–167. https://doi.org/10.1145/2090116.2090142
21. Parry, D., Davidson, B., Sewall, C., Fisher, J., Mieczkowski, H., & Quintana, D. (2021). A systematic review and meta-analysis of discrepancies between logged and self-reported digital media use. *Nature Human Behaviour*. https://doi.org/10.1038/s41562-021-01117-5
22. Rosen, J. A., Porter, S. R., & Rogers, J. (2017). Understanding Student Self-Reports of Academic Performance and Course-Taking Behavior. *AERA Open*, *3*(2), 2332858417711427. https://doi.org/10.1177/2332858417711427
23. Santos, J. L., Govaerts, S., Verbert, K., & Duval, E. (2012). Goal-oriented visualizations of activity tracking: A case study with engineering students. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, 143–152. https://doi.org/10.1145/2330601.2330639
24. Sheshadri, A., Gitinabard, N., Lynch, C. F., Barnes, T., & Heckman, S. (2019). *Predicting Student Performance Based on Online Study Habits: A Study of Blended Courses*. https://arxiv.org/abs/1904.07331v1
25. Stallings, J. (1980). Allocated Academic Learning Time Revisited, or Beyond Time on Task. *Educational Researcher*, *9*(11), 11–16. https://doi.org/10.3102/0013189X009011011
26. Verbert, K., Manouselis, N., Drachsler, H., & Duval, E. (2012). *Dataset-Driven Research to Support Learning and Knowledge Analytics*. 16.
27. Vicente, F. E., & Hellas, A. (2021). *Fine-Grained Versus Coarse-Grained Data for Estimating Time-on-Task in Learning Programming*. https://www.semanticscholar.org/paper/Fine-Grained-Versus-Coarse-Grained-Data-for-in-Vicente-Hellas/995932eae798f67739774d95ac8cc656d7977c75
28. Zhu, H., Chen, J., Wang, H., Almoaiqel, S., Materia, F., Wang, X., Cope, N., & Carroll, J. (2019, January 8). *Day Re-construction: Understanding How College Students Manage Their Time Through Self-monitoring*. https://doi.org/10.24251/HICSS.2019.462

9 APPENDICES

9.1 Courses

Institution	Code	Name	CUs	Start	End
Study.com	C958	Calculus I	4	13-06-2020	26-07-2020
ETS	TOEFL	English proficiency	0	06-07-2020	15-07-2020
Study.com	C175	Data Management - Foundations	3	11-07-2020	20-07-2020
Study.com	C993	Structured Query Language	4	12-07-2020	21-07-2020
Study.com	C170	Data Management - Applications	4	13-07-2020	22-07-2020
Study.com	C963	American Politics and the US Constitution	3	26-07-2020	29-07-2020
Sophia.org	C176	Business of IT - Project Management	4	29-07-2020	29-07-2020
Sophia.org	C165	Integrated Physical Sciences	3	30-07-2020	30-07-2020
Sophia.org	C955	Applied Probability and Statistics	3	31-07-2020	03-08-2020
WGU	ORA1	Orientation	0	25-08-2020	25-08-2020
WGU	C172	Network and Security - Foundations	3	01-09-2020	01-09-2020
WGU	C173	Scripting and Programming - Foundations	3	01-09-2020	01-09-2020
WGU	C182	Introduction to IT	4	01-09-2020	01-09-2020
WGU	C779	Web Development Foundations	3	01-09-2020	01-09-2020
WGU	C867	Scripting and Programming - Applications	4	01-09-2020	02-09-2020
WGU	C683	Natural Science Lab	2	02-09-2020	02-09-2020
WGU	C836	Fundamentals of Information Security	3	03-09-2020	03-09-2020
WGU	C959	Discrete Mathematics I	4	03-09-2020	03-09-2020
WGU	C482	Software I	6	04-09-2020	07-09-2020
WGU	C195	Software II - Advanced Java Concepts	6	07-09-2020	09-09-2020
WGU	C949	Data Structures and Algorithms I	4	09-09-2020	10-09-2020
WGU	C952	Computer Architecture	3	10-09-2020	12-09-2020
WGU	C961	Ethics in Technology	3	10-09-2020	10-09-2020
WGU	C191	Operating Systems for Programmers	3	13-09-2020	15-09-2020
WGU	C455	English Composition I	3	16-09-2020	21-09-2020
WGU	C960	Discrete Mathematics 2	4	19-09-2020	22-09-2020
WGU	C950	Data Structures and Algorithms II	4	21-09-2020	29-09-2020

Institution	Code	Name	CUs	Start	End
WGU	C188	Software Engineering	4	30-09-2020	01-10-2020
WGU	C846	Business of IT - Applications	4	02-10-2020	08-10-2020
WGU	C768	Technical Communication	3	08-10-2020	09-10-2020
WGU	C951	Introduction to Artificial Intelligence	3	10-10-2020	12-10-2020
WGU	C857	Software Quality Assurance	3	12-10-2020	17-10-2020
WGU	C964	Computer Science Capstone	4	19-10-2020	03-11-2020

9.2 Labels and examples

Label	Title	Application / URL
c165	Environmental Science	http://sophia.org/spcc/environmental-science-2
c170	CS303 Database Management - Assignment_ Database System.docx - WordPad	wordpad.exe
c172	Lecture OSI and TCP/IP Models	com.google.android.youtube
c173	C173 Scripting and Programming - Foundations – Completed (w/Study Notes) : WGU - Google Chrome	chrome.exe
c175	C175 or C836? : WGU_CompSci - Google Chrome	chrome.exe
c176	BUS402: Project Management ǀ Saylor Academy	http://learn.saylor.org/course/bus402
c182	C182 Introduction to IT : WGU_CompSci - Google Chrome	chrome.exe
c188	Software Solution Proposal.docx - Word	WINWORD.EXE
c191	Abridged Wiley Operating Systems.pdf - Personnel - Microsoft Edge	msedge.exe
c195	Software2 – Logger.java	idea64.exe
c455	C455 Task 4 Proposal Essay.docx - Word	WINWORD.EXE
c482	Software 1 GUI Mock-Up.docx - Protected View - Word	WINWORD.EXE
c683	C683 Natural Science Lab - Task 1.docx - Word	WINWORD.EXE
c768	C768 Technical Communication Task 2.docx - Word	WINWORD.EXE
c779	ciw certification - Google Search - Google Chrome	chrome.exe
c836	C836 Fundamentals of Information Security Coaching Report	http://access.wgu.edu/ASP3/aap/content/c836_security_concepts.html
c846	ITIL 4 Foundation Exam Preparation: 40 Practice Questions (	http://my.wgu.edu/courses/cour

Label	Title	Application / URL
	Part -01)	se/16000014
c857	C857 Software Quality Assurance - Passed - WGU_CompSci	http://reddit.com/r/WGU_CompSci/comments/is1w1p/c857_software_quality_assurance_passed/
c867	ClassRoster - Microsoft Visual Studio	devenv.exe
c949	New Course Instructor! C949 - mroch42@wgu.edu - Western Governors University Mail - Google Chrome	chrome.exe
c950	WGUPS Package File.xlsx [Protected View] - Excel (Unlicensed Product)	EXCEL.EXE
c951	CoppeliaSim Edu - New file - rendering: 11 ms (0.61 fps) - SIMULATION STOPPED	coppeliaSim.exe
c952	C952_ Computer Architecture home.pdf - Personal - Microsoft Edge	msedge.exe
c955	Probability Part 1: Rules and Patterns: Crash Course Statistics #13	com.android.chrome
c958	Calculus_Cheat_Sheet_All.pdf - Personal - Microsoft Edge	msedge.exe
c959	Needing Encouragement - General Advice - Discrete Math I C959 & Want to finish BSCS in 2 terms : WGU_CompSci - Google Chrome	chrome.exe
c960	C960 Notes and Links.docx - Protected View - Word	WINWORD.EXE
c961	Ethics Defined: Virtue Ethics	http://my.wgu.edu/coaching-report/v4-1/studentPidm/1401019/assessmentId/7480019/courseVersionId/12620005/assessmentCode/PGWO/
c963	Introduction: Crash Course U.S. Government and Politics	com.google.android.youtube
c964	Computer Science Capstone Topic Approval Form.docx - WordPad	wordpad.exe
c993	MySQL Installer	MySQLInstaller.exe
cs	Ep 118 Five scientific steps to ace your next exam	com.google.android.googlequicksearchbox
ora1	GOM1 - TASK 1 - ORIENTATION.docx - Word	WINWORD.EXE
other	Setup - Sweet Home 3D	SweetHome3D-6.3-windows (1).tmp
sophia.org	Sophia :: Welcome	http://sophia.org/home
study.com	Study.com - Search for courses, lessons and more - Google Chrome	chrome.exe
toefl	ETS Online Test	rpLauncherMain.exe
wgu	OneDrive - Western Governors University	explorer.exe

9.3 Learning Management System domains

Institution or Partner	Domain
ETS	go.proctoru.com
ETS	support.proctoru.com
ETS	v2.ereg.ets.org
ETS	www.ets.org
Sophia Learning	sophia.org
Sophia Learning	wgu.sophia.org
Sophia Learning	www.sophia.org
Study.com	study.com
Study.com	support.study.com
WGU	access.wgu.edu
WGU	asa.wgu.edu
WGU	cm.wgu.edu
WGU	inquiry.wgu.edu
WGU	learn.zybooks.com
WGU	lrps.wgu.edu
WGU	my.wgu.edu
WGU	partners.wgu.edu
WGU	secure.wguassessment.com
WGU	srm--c.na127.content.force.com
WGU	srm--c.na127.visual.force.com
WGU	tasks.wgu.edu
WGU	vsa.wgu.edu
WGU	web5.wgu.edu
WGU	westerngovernorsuniversity-my.sharepoint.com
WGU	wgu-nx.acrobatiq.com
WGU	wgu.edu
WGU	wgu.examity.com
WGU	wgu.hosted.panopto.com

Institution or Partner	Domain
ETS	go.proctoru.com
ETS	support.proctoru.com
ETS	v2.ereg.ets.org
ETS	www.ets.org
Sophia Learning	sophia.org
Sophia Learning	wgu.sophia.org
Sophia Learning	www.sophia.org
Study.com	study.com
Study.com	support.study.com
WGU	access.wgu.edu
WGU	wgu.mindedgeonline.com
WGU	wgu.ucertify.com
WGU	wgu.webex.com
WGU	www.ucertify.com

9.4 Most used learning resources