Koh, E., et al. (Eds.) (2026). Proceedings of the 2nd International Conference on Learning Evidence and Analytics. Asia-Pacific Society for Computers in Education.

Event Log Analysis of In-Class Assignments Using Jupyter Notebook and the Moodle Quiz Module

Nobukuni HAMAMOTO[a*], Shigetoshi YOKOYAMA[b] Atsuko, TAKEFUSA[b] & Kento AIDA[b]
[a]*Library and IT Center, Gunma University, Japan*
[b]*National Institute of Informatics, Japan*
*n.hamamoto@gunma-u.ac.jp

Abstract: Although CoursewareHub enables Jupyter Notebook log acquisition, established analysis methods are lacking. This study classifies students by treating their behavioral data as event sequences. Using Levenshtein, Normalized Levenshtein, and DTW distances for hierarchical clustering, we analyzed correlations with Quiz scores of Moodle. Findings indicate that high-performing students exhibit specific patterns: approximately 40 events with repeated cell executions, longer deliberation times, and immediate LMS access at the lecture's start.
Keywords: Jupyter Notebook, Moodle, Learning Log Analysis

1. Introduction

While Jupyter Notebook is increasingly used for programming and active learning, its local setup (e.g., via Anaconda) remains a challenge for the first-year students of the university due to their limited IT literacy. Google Colaboratory offers configuration-free access but hinders progress monitoring as instructors cannot view student data or execution logs on Google's servers. To address this, the National Institute of Informatics provides CoursewareHub, a JupyterHub-based extension (Nagaku, et al., 2019; NII Cloud Operation Team, n.d.). It supports lecture-specific functions like user authentication and LTI integration. Importantly, storing execution logs on the server offers the potential for remote monitoring of student progress.

Although CoursewareHub has been adopted for programming courses at Muroran Institute of Technology (Kuwata, et al., 2020) and Gunma University (Hamamoto et al., 2024, 2025), log analysis methods remain unestablished. By integrating Moodle logs and Notebook execution sequences, this study aims to identify specific behaviors, such as repeated cell executions, which could not be detected by previous analyses (Hamamoto et al., 2024). We examine the relationship between real-time execution status and Moodle assignment scores to better understand student progress.

2. Course Design

This paper focuses on the liberal arts course "Computer Networks and Security" offered at Gunma University, primarily for first-year students. Throughout 10 of the 15 lectures, students used Jupyter Notebook to complete Moodle-based assignments (Feedback, Quiz, and Assignment modules). Notebook was used in 10 out of the 15 lectures. We specifically examine Lecture 13, which utilized the Quiz module to cover the RSA cryptosystem. This topic is well-suited for Notebook due to the modular arithmetic required for RSA. The quiz comprised the following three questions: (1) When $p = 108$, $q = 431$, and $e = 11$, what is the smallest value of d? (2) What is the ciphertext when encrypting $M = 12345$? (3) What is the signing text when signing $M = 6789$? Students answered questions using the Notebook described in ref (Hamamoto, n.d.).

The lecture is structured as follows. (1) The teacher explains RSA cryptography. This includes demonstrating the ciphertext $C \equiv Me \pmod{n}$ for plaintext M and the decrypted text $M' \equiv Cd \pmod{n}$ explaining Euler's Theorem and the selection of $e, n,$ and d based on it. (2)

Simple problems, like those in Moodle quizzes, are presented. The teacher solves these problems using Notebook. (3) Moodle quizzes are presented, and students solve them using Notebook.

3. Methods

This paper analyzes Moodle and CoursewareHub cell execution logs to classify student behavior and examine its relationship with initial quiz scores. While Hamamoto et al (2024) found no correlation between scores and features such as total or per-cell executions, this study focuses on event sequences occurring before the first submission. Two event types are defined: (1) Notebook cell executions and (2) Moodle-specific activities. Student classification is performed by comparing three distance metrics: Levenshtein Distance (LD), Normalized Levenshtein Distance (NLD), and Dynamic Time Warping (DTW).

3.1 Levenshtein Distance (LD)

The Levenshtein distance (LD) measures the minimum operations (insertion, deletion, or substitution) required to transform one string into another (Levenshtein,1966). We treat each student's chronological event sequence as a string, which is used as a feature of the student. LD is employed as the distance between strings.

The characters assigned to events are defined as follows. For “Event Type 1,” which represents each student's cell execution, assign a 20-character code based on the cell number. The cells relevant to this quiz are those with IDs 6 to 21 and 23 to 26. The character string is assigned in ascending ID order using lowercase letters a, b, …, t. For “Event Type 2,” assign the codes shown in Table 1 to each Moodle event name. For example, if events occur in the following sequence for a user: “Cell 6 execution (a), Quiz taken (B), Cell 8 execution (c),” the user's feature is represented by the string “aBc.” The Levenshtein distance is calculated using library (python-Levenshtein, n.d.). Classification employs hierarchical clustering.

Table 1 *Event name of LMS and assigned code*

Event Name	Code	Event Name	Code
Course module viewed	A	Quiz report viewed	E
Quiz attempt viewed	B	Quiz attempt reviewed	F
Quiz attempt started	C	Course activity completion updated	G
Quiz attempt updated	D	Quiz attempt submitted	H

3.2 Normalized Levenshtein Distance (NLD)

The Levenshtein distance is sensitive to differences in string length, hence a normalized version of the Levenshtein distance relative to length has been proposed. Here, we adopt the following definition for the normalized Levenshtein distance: $\frac{2L(ab)}{N(a)+N(b)+L(ab)}$, where $L(ab)$ and $N(a)$ are Levenshtein distance and length respectively. This distance has been proven to satisfy the axioms of distance (Li, Liu, 2007).

3.3 Dynamic Time Warping Distance Measure (DTW)

In string-based representations of event sequences, time intervals are not considered and each event is treated independently, failing to reflect the sequential nature of notebook execution. To address this, we assigned an order to the events and performed classification using similarity measures based on Dynamic Time Warping (DTW) (Salvador, Chan, 2007; Fastdtw, n.d.). The sequence of events was arranged in the following order, following the expected execution sequence of events during the lecture progression: ACBabcdefghijklmnopqrstDEGHF. “A” was assigned the value 1, and the values were incremented by 1 up to “H”. In other words, if we denote the numerical value corresponding to an event by n, then $n(A) = 1, n(C) = 2, \dots n(H) = 28$.

4. Results and Discussions

Student event time series are used as features. After calculating Levenshtein distance, normalized Levenshtein distance, and DTW distance, hierarchical clustering is performed for classification. To understand the characteristics of each group, we calculate the length of each event sequence. Table 2 shows the mean and standard deviation of the event sequence length for each group.

Table 2. *Average and standard deviation of the length of event execution sequence*

	LD					NLD					DTW				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
mean	17.0	26.6	53.5	35.0	44.8	17.0	26.0	43.3	44.2	27.4	26.9	30.6	35.5	34.9	38.7
std	2.9	3.2	9.2	3.2	4.7	2.9	3.2	6.1	10.4	3.3	11.2	10.5	7.8	9.8	14.6

4.1 Characteristics of Classification Results

As shown in Table 2, LD-based classification correlates strongly with event sequence length, evidenced by distinct mean lengths and small standard deviations across groups. Conversely, NLD yields groups (e.g., c/d and b/e) with similar average lengths, suggesting that distance alone is less discriminative than in LD. Finally, DTW classification appears largely length-independent, characterized by similar mean lengths and large standard deviations across all groups.

Table 3 *The relationship between classification results and problem scores*

		LD					NLD					DTW				
		a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
Score	0	0	1	0	1	1	0	0	2	0	1	0	3	0	0	0
	3.33	1	4	0	0	0	1	4	0	0	0	2	2	0	0	1
	6.67	2	3	1	2	0	2	0	1	2	3	3	3	0	1	1
	10	3	10	1	1	10	3	6	9	3	4	3	9	2	6	5
	ave.	7.8	7.4	8.3	5.8	9.1	7.8	7.3	8.1	8.7	7.5	7.1	6.9	10	9.5	8.6

4.2 Relation with Score

Table 3 shows the relationship between classifications and scores. High-scoring groups (averaging >8 points) include: LD (Groups c, e), NLD (c, d), and DTW (c, d, e). Among these, the largest groups are LD Group e (11 members), NLD Group c (12 members), and DTW Groups d and e (7 members each). Regardless of the classification method employed, no groups consisting exclusively of low-achieving students were identified.

High-scoring LD groups (c and e) exhibit longer event sequences than others. Group e (mean length: 44.8) typically repeats the entire sequence twice, while Group c (mean length: 53.5) shows frequent repetitions of quiz-related cells. Although Group c contains only two members, these results suggest that high scores correlate with approximately 40 event executions and repeated cell engagement.

LD Group d was expected to achieve high scores because its mean sequence length of 35 was close to the 40-execution pattern. However, the group’s average score remained low. This was due to the small sample size (n=4), where the mean was distorted by a single student who scored zero. In fact, the other three members all achieved high scores.

NLD Groups c and d also achieved high average scores, with mean sequence lengths of 43.3 and 44.2, respectively. These results confirm that the 40-event threshold remains beneficial in NLD, inheriting the repeated execution patterns observed in LD Group c. Although NLD effectively distinguishes behavioral patterns among groups with similar sequence lengths (e.g., b/e and c/d), no significant score differences were observed between them.

Unlike LD or NLD, DTW Groups c, d, and e are primarily influenced by execution timing rather than sequence length. High-scoring Group e is characterized by immediate LMS access following the instructor's introduction, suggesting high attentiveness. Group d features longer execution times than Groups a and b, possibly indicating deeper engagement or reflection. Group c exhibits slower execution, though it remains unclear whether this reflects independent deliberation or peer consultation following delayed starts.

In summary, higher scores correlate with slower execution times and prompt LMS access following instructor introduction.

5. Summary and Conclusions

This paper details the implementation of educational practices using CoursewareHub and Moodle. By analysing notebook materials and Moodle Quiz logs, classifying them using event sequences as features and applying Levenshtein distance, normalised Levenshtein distance, and DTW distance, students exhibiting favourable learning behaviours were identified across multiple groups. Verification of each tendency revealed the following patterns as potential indicators for identifying high-achieving students:

(1) Approximately 40 event length with repeated cell execution.
(2) Completion is not achieved early.
(3) The LMS is executed at the start of the lecture.

The results obtained thus far capture the characteristics of high-scoring students. For future work, we aim to develop this into predictive outcomes based on trends observed during lecture progression.

Acknowledgements

In this research work, we used the “mdx: a platform for building data-empowered society.” This work was supported by JSPS KAKENHI Grant Number JP23K11363. We would like to thank Mr. Atsunobu KOIZUMI for supporting the construction of the CoursewareHub environment on mdx.

References

Nagaku M., Masatani M. & Aida K. (2019). CoursewareHub: A classroom tool for conducting lessons based on Notebooks, *IPSJ SIG Technical Report*, 2019-CLE-27(20),1–4.

NII Cloud Operation Team (n.d.). ocstemplates/CoursewareHub, https://github.com/nii-gakunin-cloud/ocs-templates.

Kuwata Y., Ishizaka T., Ogawa Y., Masatani Y., Nagaku M., Yokoyama S., & Hamamoto N. (2020). Knowing the Progress of Programming Exercises by make use of the History of Jupyter Notebook Environment. *JSAI SIG Technical Report*, 2020(KSN-024-04), 1–6.

Hamamoto, N., Ogawa, K., Yokoyama, S., Takefusa, A. & Aida, K. (2024). The use of CoursewareHub in Moodle quizzes on the subject of public key cryptography and its log analysis, *IPSJ SIG Technical Report*, 2024-CLE-44(3), 1–8.

Hamamoto, N., Ogawa, K., Yokoyama, S., Takefusa, A. & Aida, K. (2025). The Log Analysis Environment to Supoort Classroom Using CoursewareHub, *Procedia Computer Science* 270, 3431-3439.

Hamamoto, N (n.d.) https://github.com/n-hamamoto/nwsec2023/blob/main/TBICS/13_RSA.ipynb.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, 10(8), 707–710.

python-Levenshtein (n. d.) https://pypi.org/project/python-Levenshtein/.

Li Y& Liu B. (2007). A Normalized Levenshtein Distance Metric. *IEEE transactions on pattern analysis and machine intelligence*. 29. 1091-5. 10.1109/TPAMI.2007.1078.

Salvador S., & Chan P. (2007) FastDTW: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11(5) 561-580.

Fastdtw (n.d.), https://pypi.org/project/fastdtw/.