Mining the Sequential Dependencies in Learning Events with a Self-Attentive Hawkes Process

Sijing Yu
South China Normal University
yusijing@m.scnu.edu.cn

Tianwei Peng
South China Normal University
2024020894@m.scnu.edu.cn

Sijie Zhang
South China Normal University
2024020913@m.scnu.edu.cn

Zengcan Xue
South China Normal University
xuezc@m.scnu.edu.cn

Su Mu
South China Normal University
musu@m.scnu.edu.cn

Yingbin Zhang
South China Normal University
zyingbin@m.scnu.edu.cn

ABSTRACT
Modern educational settings routinely generate massive amounts of learning event sequences, which often exhibit complicated temporal and sequential dependencies among events. Traditional sequence dependency analysis methods, such as sequential pattern mining, requires researchers to manually configure pattern parameters, such like minimum support, positional gap, and time interval, which introduces subjectivity into the analysis process. Meanwhile, deep learning models combined with temporal point processes, particularly Hawkes processes with attention mechanisms, have shown promise in capturing sequential dependencies and predicting future events, yet existing models remain limited in the interpretability of the dependency features they identify. To address these limitations, this study proposed a method based on self-attentive Hawkes processes to identify the sequential dependencies in learning event sequences without requiring manual configuration of pattern parameters. We applied the proposed method to two learning event datasets and compared the extracted patterns with those mined by two baseline algorithms. The results revealed a relatively small degree of pattern overlap between the proposed method and baselines. For overlapping patterns, the proposed method identified temporal characteristics differed significantly from that of baselines, which were manually preset.

Keywords
Learning event dependency, sequential pattern mining, temporal pattern mining

1. INTRODUCTION
Learning process data are typically represented as sequences of learning events that partially reflect learners' cognitive processes and behavioral patterns. The temporal sequential relationships among learning events, referred to as sequential dependencies, hold significant pedagogical value [1]. They can support a wide range of educational applications, such as learning performance prediction and learning environment design. For example, sequential dependencies in MOOC learning events can serve as features to predict students' dropout risk [2]; that in video viewing behaviors can help reveal the engagement with instructional videos [3]; and that in the resource usage behaviors among a certain group of learners can be transformed into learning paths recommended to other learners in the same group [4].

A range of methods has been used to analyze sequence dependencies in learning events, include sequential pattern mining (SPM), lag sequential analysis, process mining, and ordered network analysis [5]. However, except for SPM, these methods are generally limited to characterizing pairwise dependencies between events and have difficulty capturing the overall sequential dependencies among multiple learning events. SPM can mine sequence patterns of various lengths [6] and has been widely used in descriptive, relational, and predictive studies [7, 8, 9], such as mining learning behaviors, generating features for prediction and classification models, and filtering learning resources to build recommendation systems. However, it relies on researchers' manual setup of pattern parameters, such as the minimum support (the minimum sequences containing a pattern) as well as positional gaps and time intervals among adjacent pattern events. Because these preset parameters inevitably involve subjective judgment, the manual setup creates a gray area in the analytical process that may introduce bias into the results [10], thereby influencing researchers' understanding of learning patterns.

Recent advances in deep learning have combined deep neural networks with temporal point process models to characterize sequential dependencies while taking into account factors such as event position and time interval, thereby enabling future event prediction [11]. Among these models, Hawkes-process-based approaches are particularly suitable because they capture how prior events influence subsequent events over time. Accordingly, Hawkes processes have been widely applied to event prediction tasks, such as predicting procrastination in online learning [12]. More recently, attention-based extensions such as the self-attentive Hawkes process (SAHP) [13] and the Transformer Hawkes Process [14] have further improved the modeling of event sequences and outperformed recurrent neural network-based Hawkes process models in future event prediction. However, these models are primarily designed for event prediction and extrinsic target classification, which limits their usefulness for extracting interpretable sequential dependency features.

To address these limitations, this study proposes a novel method that combines the self-attentive Hawkes process with the task of mining learning event sequential dependencies. The proposed metho aims to reduce reliance on manually preset pattern parameters while extract interpretable patterns of learning events with relatively strong sequential dependency. To evaluate its effectiveness in relation to traditional sequence dependency analysis, this study compares the patterns and time intervals identified by the proposed method with those obtained from baseline SPM algorithms. Specifically, this study examines the following research questions:

- RQ1: To what extent do the pattern sets mined by the proposed method overlap with those mined by baseline SPM algorithms?
- RQ2: For overlapping patterns, how do the time intervals automatically identified by the proposed method compare with those preset in baseline SPM algorithms?

2. METHODOLOGY

We refer to the proposed method as the leSPM-SAHP algorithm. It builds on the SAHP architecture [13] because SAHP showed high performance in modeling event dependencies and event prediction while incorporating temporal order and time intervals, making it suitable for learning-event sequence analysis. Specifically, in self-supervised learning where the task is the next event prediction, the model uses the attention output O, derived from the query (Q), key (K), and value (V) matrices, to quantify the influence of historical events on subsequent events.

$$O = AV \quad A = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{M_K}}\right)$$

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

The value a_{ij} in the i-th row and j-th column of the attention weight matrix A represents the strength of sequential dependencies of event (k_i, t_i) on historical event (k_j, t_j) [14]. k_i is the event type of the i-th event, while t_i is the time elapsed since the first event. To prevent overly distant events from contributing noise, we introduced a history-length regularization parameter ℓ, which restricts attention to a bounded temporal context by forcing attention weights beyond ℓ positions to approach zero. Correspondingly, if $i - j > \ell$, a_{ij} approaches 0.

Based on the resulting attention weights, leSPM-SAHP first identifies candidate pairwise sequential dependencies between events. For each event pair, a Wilcoxon signed-rank test is used to determine whether its attention weights are significantly greater than the baseline threshold ($\frac{1}{\ell+1}$), so that only statistically significantly dependencies are retained.

Because pairwise dependencies correspond only to patterns of two events, we then extend them into longer patterns through a layer-linking pattern expansion procedure. This procedure is motivated by the ability of higher-order attention layers to encode dependencies involving multiple events. By sequentially linking sequence patterns across adjacent layers, short dependencies are progressively expanded into longer event sequence patterns.

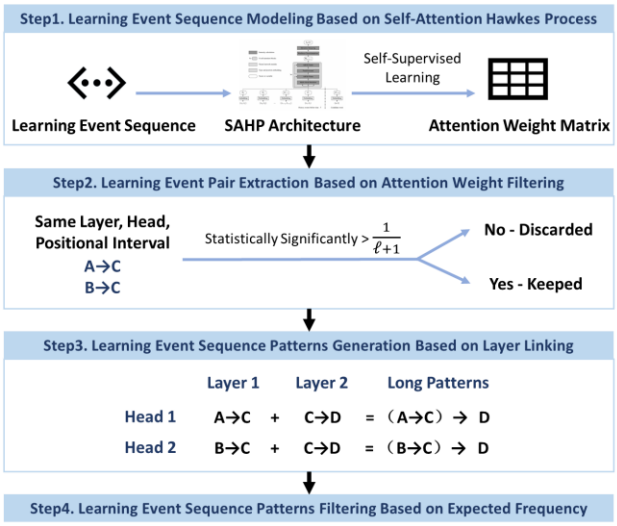

Figure 1. The workflow of leSPM-SAHP

Finally, to reduce spurious or practically uninformative outputs, all extracted patterns are filtered using an expected-frequency criterion, which were based on the product of the simple frequencies of pattern events. Patterns with occurrence frequencies lower than their expected frequency in the event sequences are removed, and the remaining patterns constitute the final pattern set. This design allows leSPM-SAHP to reduce reliance on manually preset pattern parameters in traditional SPM while preserving interpretable pattern and time-interval information for comparison with baseline SPM algorithms.

3. EXPERIMENT

3.1 Datasets

We evaluated leSPM-SAHP on two contrasting learning-event datasets, a large-scale *XuetangX MOOC*[1] dataset and an event-rich middle-school science game dataset, *Wake: Tales from the Aqualab*[2], to examine whether the method performs consistently across different learning contexts. Table 1 displays basic statistics of the two datasets. To reduce excessively long inactive gaps, sequences in both datasets were split whenever the interval between adjacent events exceeded 15 minutes.

Table 1. Basic Information of Datasets

Dataset	Student Type	Number of Students	Mean Sequence Length (*SD*)	Number of Event Type
MOOC	Multiple Types	6343	157.65 (*568.51*)	21
Aqualab	Middle School Students	590	817.01 (*1492.10*)	58

[1] http://moocdata.cn/data/user-activity

[2] https://opengamedata.fielddaylab.wisc.edu/gamedata.php?game=AQUALAB

3.2 Baselines

We compared leSPM-SAHP with two recent temporal pattern mining baselines, both of which support time-interval sequential patterns extraction:

- **OER-Miner (2025):** This method employed episode connection strategies and position index support calculation to efficiently mines sequential patterns with time interval constraints [15].
- **TIRPClo (2023):** It achieves complete temporal interval pattern mining through entity projection based on endpoint sequence transformation and memory indexing [16].

3.3 Analysis

For leSPM-SAHP, model hyperparameters were selected using 5-fold cross-validation, and attention-weight filtering used a significance level of 0.05. For the baselines, pattern sets were extracted under multiple combinations of minimum support and maximum time-interval settings to provide a systematic comparison across parameter conditions (Table 2). A pattern was returned by the baseline only when the proportion of sequences with at least one valid pattern instance was no less than the minimum support, and a pattern instance was valid when the time interval between adjacent events was not greater than the maximum time interval.

Table 2. Settings of Pattern Parameters. 25th TI denotes the time interval at the 25th percentile rank among all event time intervals (the time interval between the first event of a sequence and all other events in the sequence)

Dataset	Parameter	Level
MOOC	Minimum Support	0.15, 0.20, 0.25, 0.30
	Maximum Time Interval	25th TI, 50th TI, 75th TI
Aqualab	Minimum Support	0.55, 0.60, 0.65, 0.70
	Maximum Time Interval	25th TI, 50th TI, 75th TI

Under each combination of pattern parameters, we compared the filtered pattern set P produced by leSPM-SAHP with the baseline pattern set P_{base} (see Figure 2). Specifically, a denotes the count of patterns identified only by leSPM-SAHP, b denotes the count of patterns identified only by the baseline method, and c denotes the count of overlapping patterns. Before filtering, the corresponding counts were recorded as a', b', c'. For the overlapping patterns (c), we further compared the differences in maximum time interval identified by leSPM-SAHP with those preset in the baseline SPM algorithms.

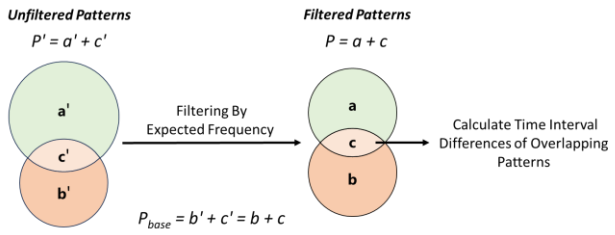

Figure 2. Metrics for comparing leSPM-SAHP with baselines. P' was the unfiltered pattern set extracted by leSPM-SAHP, while P was the filtered pattern set. P_{base} was the pattern set extracted by the baselines

We focused on three metrics of pattern overlapping: $\frac{c}{a+c}$, $\frac{b}{b+c}$, and $\frac{b\prime}{b\prime+c\prime}$. If $\frac{c}{a+c}$ is small and the interval difference is large, it means that the sequence patterns extracted by leSPM-SAHP cannot be covered by existing algorithms. If $\frac{b}{b+c}$ is large and $\frac{b\prime}{b\prime+c\prime}$ is small, it indicates that many patterns extracted by baselines existed in the unfiltered pattern set by leSPM-SAHP but not in the filtered pattern set, which means baselines may extract many patterns without relatively strong event dependencies as their frequencies lower than the expected frequency.

4. RESULTS & DISCUSSION

In the MOOC dataset, the number of patterns mined by OER-Miner ranged from 23 to 2385, that by TIRPClo ranged from 16 to 794, while that by leSPM-SAHP was 630. In the Aqualab dataset, the number of patterns mined by OER-Miner ranged from 765 to 7305, that by TIRPClo ranged from 603 to 6728, while that by leSPM-SAHP was 3653.

As shown in Figure 3, in both datasets, $\frac{c}{a+c}$ consistently remained at a low level, with mean values of 3.46% and 0.76% for MOOC and Aqualab datasets, respectively. This indicates that only a small proportion of the sequential patterns in the filtered set extracted by leSPM-SAHP overlapped with those extracted by the baseline algorithms. In other words, most patterns identified by leSPM-SAHP were not covered by the baseline under the tested parameter settings.

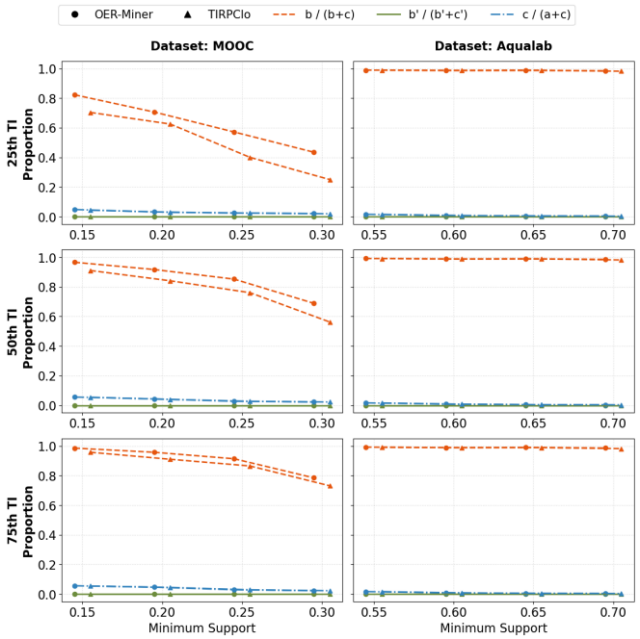

Figure 3. The three metrics under each parameter combination

More importantly, across all combinations in both datasets, $\frac{b\prime}{b\prime+c\prime}$ was 0. This indicates that the patterns extracted by the baseline algorithms were also present in the unfiltered output of leSPM-SAHP. However, in the Aqualab dataset, $\frac{b}{b+c}$ remained at a high level, fluctuating between 98.14% and 99.19%, and in the MOOC dataset, it was also large under many parameter combinations, although smaller in some conditions. Taken together, these results suggest that the divergence between leSPM-SAHP and the baseline methods arose primarily from the filtering stage rather than from a failure of leSPM-SAHP to initially identify the patterns extracted

by the baselines. In other words, many patterns extracted by the baseline algorithms existed in the unfiltered pattern set of leSPM-SAHP but were not retained after filtering. This suggests that frequency-based SPM algorithms may preserve many patterns that satisfy support-based criteria but are not retained under the dependency- and expected-frequency-based filtering procedure of leSPM-SAHP.

For overlapping patterns, we next examined whether the maximum time intervals identified by leSPM-SAHP differed significantly from those preset in the baseline SPM algorithms. For each fixed combination of dataset, maximum time interval, algorithm, and minimum support, we tested whether the time interval differences significantly differed from zero using the Wilcoxon signed-rank test. To control for error due to multiple comparisons, p-values were adjusted using the Benjamini-Yekutieli correction.

Figure 4 displays the heatmap of mean time interval differences across datasets, maximum time intervals, baseline algorithms, and minimal support. In the MOOC dataset, the time interval differences were uniformly positive across all conditions, and all 24 cells were statistically significant. This indicates that, for overlapping patterns, the maximum time intervals identified by leSPM-SAHP were consistently larger than those preset in the baseline methods.

In Aqualab, time interval differences were also consistently positive and significant under the 25th and 50th TI thresholds. By contrast, under the 75th TI threshold, all Aqualab cells turned negative, and five of the eight cells remained statistically significant after correction. Thus, while the MOOC results showed a stable pattern of larger time intervals under leSPM-SAHP, the Aqualab results suggest that the direction and magnitude of time interval differences depended on the maximum time interval condition. This set of findings addresses RQ2 by showing that, even for overlapping patterns, leSPM-SAHP and the baseline SPM algorithms often assigned different temporal spans to the same pattern.

Taken together, the results indicate that the overlap between leSPM-SAHP and the baseline SPM algorithms was limited in both datasets, and that the temporal characteristics of overlapping patterns often differed significantly. A plausible explanation is that existing SPM algorithms depend on manually preset pattern parameters, such as minimum support and maximum time interval, whereas leSPM-SAHP does not rely on such preset thresholds in the same way and instead derives dependency information from the data through the SAHP model structure and filtering procedure. From this perspective, leSPM-SAHP may capture sequential patterns and time intervals among learning events that are not easily recovered when pattern parameters must be specified in advance. More broadly, these findings suggest that sequence dependency analysis in educational data mining may benefit from approaches that infer temporal and frequency constraints adaptively from data rather than relying on researcher-defined parameter settings.

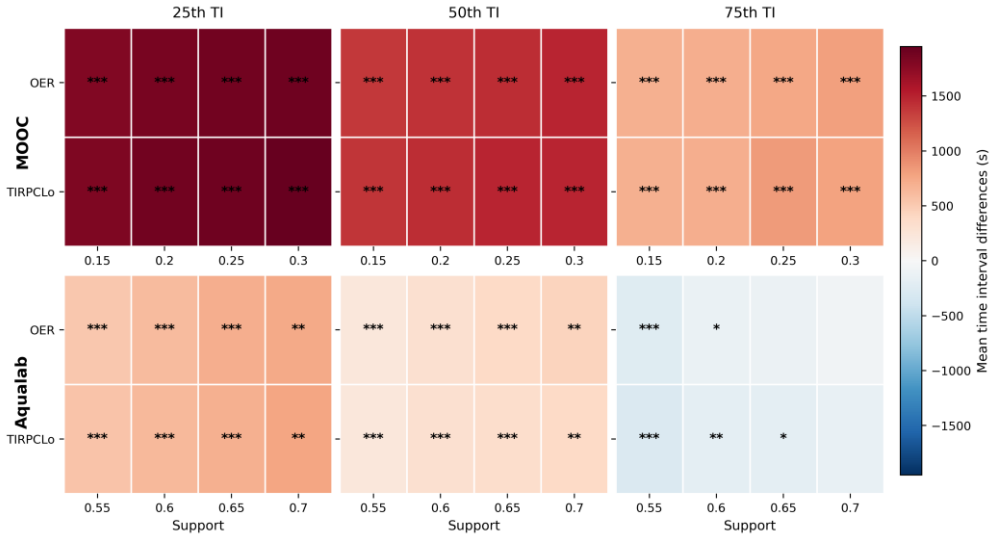

Figure 4. Comparison of time interval differences for overlapping patterns under each condition. Asterisks denote Benjamini-Yekutieli-adjusted significance against 0 (* $p < .05$, ** $p < .01$, *** $p < .001$)

5. LIMITATIONS AND FUTURE WORK

At the same time, these findings should be interpreted with caution. Low overlap does not by itself prove that leSPM-SAHP identifies more meaningful patterns than the baseline methods; rather, it shows that the proposed method yields substantially different outputs under the present comparison framework. Similarly, the fact that many baseline patterns were filtered out by leSPM-SAHP suggests that those patterns were not retained under dependency-based screening, but it does not by itself establish that they are substantively unimportant. Additional validation is therefore needed to examine whether the patterns preferentially retained by leSPM-SAHP are indeed more informative, interpretable, or pedagogically useful in real educational settings.

For overlapping patterns, this study only compared their time interval differences. Future research should further examine positional interval differences to provide a more complete comparison of pattern features across methods. Additionally, this study utilized only two datasets and two baseline algorithms. Subsequent research should extend the comparisons to data from more diverse learning contexts and a broader range of algorithms to investigate whether the degree of pattern overlap depends on dataset characteristics or algorithmic assumptions.

6. ACKNOWLEDGMENTS

This study was funded by National Natural Science Foundation of China (No.62407014).

7. REFERENCES

[1] Molenaar, I. and Wise, A. F. 2022. Temporal aspects of learning analytics: Grounding analyses in concepts of time. In *The Handbook of Learning Analytics, 2nd ed*. 66-76. DOI= https://doi.org/10.18608/hla22.006

[2] Deeva, G., De Smedt, J., and De Weerdt, J. 2022. Educational sequence mining for dropout prediction in MOOCs: Model building, evaluation, and benchmarking. *IEEE Transactions on Learning Technologies.* 15, 6 (2022), 720-735. DOI= http://doi.acm.org/10.1109/TLT.2022.3215598.

[3] Wong, J., Khalil, M., Baars, M., de Koning, B. B., and Paas, F. 2019. Exploring sequences of learner activities in relation to self-regulated learning in a massive open online course. *Computers & Education.* 140 (2019), 103595. DOI= http://doi.acm.org/10.1016/j.compedu.2019.103595.

[4] Wan, S. and Niu, Z. 2019. A hybrid e-learning recommendation approach based on learners' influence propagation. In *IEEE Transactions on Knowledge and Data Engineering*. 32, 5 (2019), 827-840.

[5] Lämsä, J., Hämäläinen, R., Koskinen, P., Viiri, J., and Lampi, E. 2021. What do we do when we analyse the temporal aspects of computer-supported collaborative learning? A systematic literature review. *Educational Research Review*. 33 (2021), 100387.

[6] Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*. 3-14. DOI= http://doi.acm.org/10.1109/icde.1995.380415.

[7] Kang, J., Liu, M., and Qu, W. 2017. Using gameplay data to examine learning behavior patterns in a serious game. *Computers in Human Behavior*. 72 (2017), 757-770. DOI= http://doi.acm.org/10.1016/j.chb.2016.09.062.

[8] Kinnebrew, J. S., Segedy, J. R., and Biswas, G. 2015. Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments. *IEEE Transactions on Learning Technologies*. 10, 2 (2015), 140-153.

[9] Slim, A., Heileman, G. L., Al-Doroubi, W., and Abdallah, C. T. 2016. The impact of course enrollment sequences on student success. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. 59-65. DOI= http://doi.acm.org/10.1109/AINA.2016.140.

[10] Zimmermann, A. 2020. Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 10, 2 (2020), e1330. DOI= http://doi.acm.org/10.1002/widm.1330.

[11] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202*. 07125 (2022). DOI= https://doi.org/10.48550/arXiv.2202.07125.

[12] Yao, M., Zhao, S., Sahebi, S., and Behnagh, R. F. 2021. Relaxed clustered Hawkes process for student procrastination modeling in MOOCs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 35, 5 (2021), 4599-4607. DOI= https://doi.org/10.1609/aaai.v35i5.16589.

[13] Zhang, Q., Lipani, A., Kirnap, O., and Yilmaz, E. 2020. Self-attentive Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*. 119 (2020), 11183-11193.

[14] Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. 2020. Transformer Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*. 119 (2020), 11692-11702.

[15] Wu, Y., Dong, Z., Liu, J., Li, Y., Liu, C., Wen, L., and Wu, X. 2025. OER-Miner: One-off episode rule mining for process event logs. *IEEE Transactions on Emerging Topics in Computing*. 13, 4 (2025), 1497-1509.

[16] Harel, O. and Moskovitch, R. 2023. TIRPClo: Efficient and complete mining of time intervals-related patterns. *Data Mining & Knowledge Discovery*. 37, 5 (2023). DOI= https://doi.org/10.1007/s10618-023-00944-6.