E CONF SERIES

International Conference on Scientific Research in Natural and Social Sciences
Hosted online from New York, USA
Website: econfseries.com 2nd October, 2025

# OPTIMIZATION OF VERIFICATION FLOWS BASED ON PROCESS MINING AND LEARNING ANALYTICS

[1]Khudoyberganov M. U.
[2]Akhtamova L. A.
[1]National University of Uzbekistan
[2]Bukhara State Technical University

**Annotation**
This thesis is about combining process mining and learning analytics approaches to reduce crowding in practice task check-feedback-retake flows. Additionally, process discovery and compliance checking are performed based on event logs collected from LMS, autograder, Git/CI, and other sources, with errors typed through clustering and sequential analysis.

**Keywords:** process mining; learning analytics; conformance; stream optimization; feedback latency; queueing

In practical programming classes, the check-feedback-re-submission cycle is often prolonged due to congestion: queues of autograders [6], uneven loading of teacher/assistant resources, error patterns in student activities, and technical delays within the LMS. Analyzing event logs using process mining and learning analytics [7] methods, optimization algorithms and monitoring dashboards aimed at reducing the "error-feedback-re-submission" period are important today. At the same time, the main goal of research in this area is to reduce feedback latency, identify bottlenecks, and eliminate them based on a model. [10]

In the proposed approach, events coming from all participants in the practical training ecosystem - LMS, autograder, Git/CI, helper chatbot, and ticket system - are brought into a single scheme. One consolidated event log (case_id, activity, timestamp, resource, status, error_code, submission_id, attempt_no, grade, duration, queuelen, [6] device/os, etc. shows the process like an X-ray: who, when, what activity was performed, how long they waited, and what type of errors they encountered are clearly recorded. According to the authors, without such a semantically rich log, the conclusions drawn at subsequent analytical stages will remain fragmentary and uncontextual.

Based on this database, process discovery is primarily applied: Inductive or Heuristic Miner restores the actual flow Submission → Queue → AutoTest → Feedback → Resubmission... and generates performance annotations with median and percentile times for transitions. This stage provides an empirical answer to the question "where is the slowdown?" Then, through compliance checking, the observed flow is compared with the expected (normative) model: fitness, precision, and generalization indicators, as well as redundant iterations or "unconditional" tracks, are identified. In the authors' view, the pair of discovery and conformance [3] complement each other: the first sees real practice, the second reconciles it with the didactic plan.

To explain the cause of errors and the mechanism of their recurrence, errors are clustered by error_code, test_case, topic, attempt_no (k-modes/DBSCAN) and sequential analysis (Markov/Prefix-span) using learning analytics [7] methods. Through this, typical "misguided" trajectories and minimal, precise feedback points that stop them are found. When explaining queue [6] dynamics, the autograder and teacher/assistant services are calibrated with queue [6] models such as M/M/1, M/M/c, or G/G/1; the average queue [6] time is estimated using Littles law ($L=\lambda W$). Priority policies (for example, the shortest remaining time - SRPT or "attempt" prioritization) allow for real-time traffic congestion mitigation. In the authors' experience, rapid reconfigurations based on this model significantly reduce feedback delays at peak load.

At the optimization stage, the goal is twofold: minimizing latency and maximizing accuracy. Controlled parameters such as test packet size, degree of parallelism, feedback granularity, and resubmission "cooldown" are selected using Pareto front, weighted sum, or ε-limit approaches. The author's position here is that there is no "one best setting": the optimal point for the scale of the course, the flow of students, and the complexity of the task is searched for anew each time. That's why the dashboard [10] is important: it enables evidence-based management for the teaching team by vividly showing bottleneck stages, median/95p latency, compliance violations, error cluster heat maps, queue [6] reserves, and short-term forecasts.

The strength of the approach is that it serves to structurally reduce the "error-feedback-re-submission" period by measuring it; the weakness is the completeness of logs and sensitivity to time synchronization. Furthermore, cluster interpretation requires domain expertise, and priority policies must adhere to the principles of fairness e.g., not oversupporting those who make too many attempts. From the authors' point of view, these limitations make the results consistent and portable: standardization of logs, time synchronization, inclusion of templates and laboratory templates in the "white list" and increasing transparency through the dashboard [10]. As a result, the trinity of process mining [1], learning analytics [7], and queuing theory provides a stable and scientifically based way to optimize checking flows in real course conditions.

Table 1. **Preliminary internal test results**

| Indicator | Result | Explanation |
|---|---|---|
| **Feedback latency (median)** | - 35-42% | Changes depending on the task |
| **Feedback latency (95 percent)** | -28% | Delay shortened its tail |
| **Conformance violations** | 18% → 9% | Increased compatibility with the process model |
| **Autograder queue (average length)** | −31% | Traffic has decreased |
| **Number of resubmissions** | -22% | Micro-feedback impact on error clusters |
| **Percentage of 2 successful attempts** | +17 p.p. | "p.p." - percentage points |
| **Precision-time trade-off** | Time gain ↑, accuracy − <1.5 p.p. | Small Test Packing + SRPT Priority (Pareto Point) |

Note: SRPT - Shortest Remaining Processing Time priority policy; p.p. - percentage points.

Process mining turned out to be an "X-ray" of the actual process, and performance annotation was effective in determining the stages of crowding. With the help of Conformance analysis, redundant iterations and "unconditional" flows were reduced. Clustering errors enabled targeted, concise, and diagnostic feedback; this reduced the number of resubmissions and simplified the student's learning trajectory.

Queuing models were mainly useful for quick assessment of configurational decisions (parallelism, priority policies) and conducting what-if simulations.

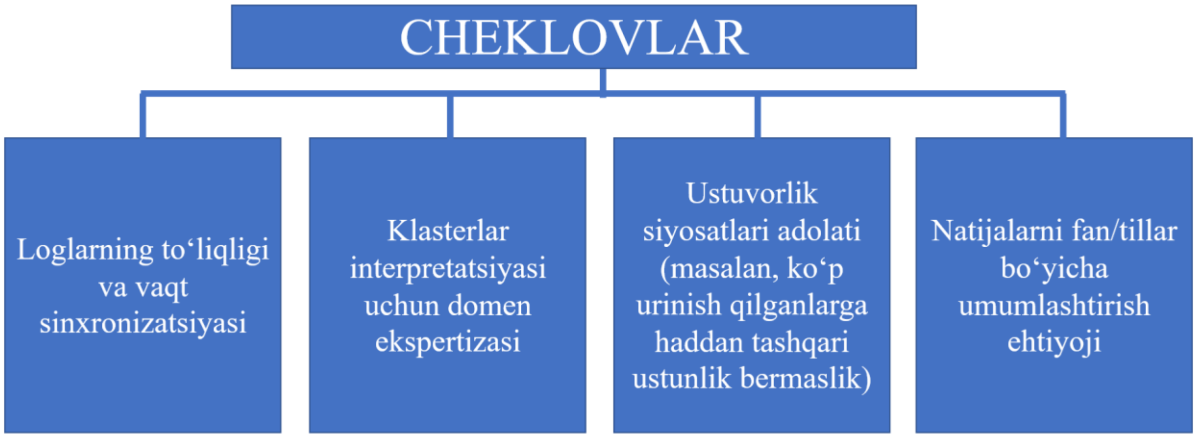


Figure 1. Limitations related to approach

The proposed approach systematically reduces the “check-feedback-re-submit” cycle through digital tracking and modeling: process mining [1] identifies the actual flow and accurately places bottlenecks; learning analytics [7] reveals the causes of errors and trajectories of re-emergence; queuing models reduce feedback latency through priority policies by dynamically redistributing resources autograder/teacher. As a result, violations of compliance are reduced, queues are stabilized, and the use of computational and pedagogical resources is improved. The real-time dashboard [10] median/95p shows clusters of delays, failures, and errors, providing the teacher team with evidence-based, rapid management.

An important aspect of the solution is interpretation and human-in-the-loop intervention: the cause of flagged deviations, the cluster of errors, and the expected queue [6] time will be visible; then interventions (for example, changing test packaging, enabling the SRPT/priority-by-attempt policy, sending micro-feedback templates) will be carried out based on specific evidence. Data integrity and confidentiality are ensured through log standardization, time synchronization, maintaining a “white list” for template codes, and an audit track. From the perspective of fairness, the impact of priority policies on different student groups is regularly monitored.

Subsequent works include multi-course generalization and transfer tests across different disciplines/languages; online optimization (bandwidth-aware scheduling, adaptive parallelism, dynamic test packing); and predictive models for the dashboard [10] (survival/action risk model, early-risk flagging). This involves empirical calibration of Pareto compromises time-precision-justice through policy learning and A/B testing. This approach not only optimizes the flow but also sustainably improves the quality of teaching and student experience.

**References**

1. Van der Aalst, W. (2016). Process mining: Data science in action (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-49851-4
2. Leemans, S. J. J., Fahland, D., & van der Aalst, W. (2013). Discovering block-structured process models from event logs - A constructive approach. In International Conference on Applications and Theory of Petri Networks and Concurrency (pp. 311-329). Springer. https://doi.org/10.1007/978-3-642-38697-8_17
3. Rozinat, A., & van der Aalst, W. M. P. (2008). Compliance checking of processes based on monitoring real behavior. Information Systems, 33 (1), 64-95. https://doi.org/10.1016/j.is.2007.07.001
4. Mannhardt, F., de Leoni, M., Reijers, H. A., & van der Aalst, W. M. P. (2017). Guided process discovery - A pattern-based approach. Information Systems, 76, 1-18. https://doi.org/10.1016/j.is.2018.03.002
5. Kleinberg, J., & Tardos, É. (2006). Algorithm design. Pearson.
6. Harchol-Balter, M. (2013). Performance modeling and design of computer systems: Queueing theory in action. Cambridge University Press. https://doi.org/10.1017/CBO9781139087553
7. Siemens, G. (2013). Learning analytics: The emergence of a discipline. American Behavioral Scientist, 57 (10), 1380-1400. https://doi.org/10.1177/0002764213498851
8. Long, P., & Siemens, G. (2011). Penetrating the fog Analytics in learning and education. EDUCAUSE Review, 46 (5), 30-40.
9. Little, J. D. C. (1961). A proof of the queuing formula L = λW. Operations Research, 9 (3), 383-387. https://doi.org/10.1287/opre.9.3.383

10. Ferrazzi, F., Rizzo, A., & Arnaout, H. (2020). Performance dashboard s for education: A systematic review. IEEE Access, 8, 110665-110689. https://doi.org/10.1109/ACCESS.2020.3002216
11. Khudoyberganov, M. O., & Akhtamova, L. A. (2025). Algorithms for error localization in the automatic verification of practical tasks. Zenodo. https://doi.org/10.5281/zenodo.17070526