LAOps: Learning Analytics with Privacy-aware MLOps

Pia Niemelä[a], Bilhanan Silverajan[b], Mikko Nurminen[c], Jenni Hukkanen[d] and Hannu-Matti Järvinen[e]

Faculty of Information Technology and Communication Sciences, Tampere University, P.O. Box 1001 FI-33014, Tampere, Finland

Keywords: Learning Management System, Next-generation Learning Environment, Assessment and Feedback, Learning Analytics, Personalisation, Machine Learning, Privacy-aware Machine Learning, Cloud-based Learning Analysis, MLOps, LAOps.

Abstract: The intake of computer science faculty has rapidly increased with simultaneous reductions to course personnel. Presently, the economy is recovering slightly, and students are entering the working life already during their studies. These reasons have fortified demands for flexibility to keep the target graduation time the same as before, even shorten it. Required flexibility is created by increasing distance learning and MOOCs, which challenges students' self-regulation skills. Teaching methods and systems need to evolve to support students' progress. At the curriculum design level, such learning analytics tools have already been taken into use. This position paper outlines a next-generation, course-scope analytics tool that utilises data from both the learning management system and Gitlab, which works here as a channel of student submissions. Gitlab provides GitOps, and GitOps will be enhanced with machine learning, thereby transforming as MLOps. MLOps that performs learning analytics, is called here LAOps. For analysis, data is copied to the cloud, and for that, it must be properly protected, after which models are trained and analyses performed. The results are provided to both teachers and students and utilised for personalisation and differentiation of exercises based on students' skill level.

1 INTRODUCTION

Learning management systems (LMSs) have grown in prominence in course management, practising and performance evaluation. In Tampere University, for example, the LMS system auto-tests and grades students' submissions, including the bigger coursework assignments. The grading data in software courses may comprise, e.g., unit/integration test reports and static code analysis. In total, the pipeline of auto-tests generates an excessive amount of data together with the data gathered from code commits and the project management tool. In essence, Tampere University already utilises student data and learning analytics for tutoring and smoothing the transfer from higher education to working life (Okkonen et al., 2020b; Heikkilä and Okkonen, 2021; Okkonen et al., 2020a), as a broader-scope target than the one of this research.

Instead of using proprietary scripts for manipulating and analyzing e-Learning data, the pipeline should be smartened up with Data/MLOps to make the analysis automatic, traceable, observable and secure. This way analysis remains well-documented and results are readily available for all project participants, and for them only by a guarantee. During the analysis phase, the data will be transferred into the cloud. The solution architecture targets bridging the learning analytics development and the identified best practices of secure data storage and handling.

LMS platforms rely heavily on cloud-based storage for user data pertaining to each user's learning patterns, personal data, results obtained and interactions. The cybersecurity aspect of this paper is to maximise and fortify the trust of users in cloud-based LMS services by developing mechanisms for protecting both derived and raw personal data. The paper then offers a novel solution through which both users and machine learning algorithms can calculate statistics or operate on data in a privacy-preserving way.

This paper is organised as follows. Chapter 2

[a] https://orcid.org/0000-0002-8673-9089
[b] https://orcid.org/0000-0001-6565-8776
[c] https://orcid.org/0000-0001-7609-8348
[d] https://orcid.org/0000-0002-7691-5974
[e] https://orcid.org/0000-0003-0047-2051

Niemelä, P., Silverajan, B., Nurminen, M., Hukkanen, J. and Järvinen, H.
LAOps: Learning Analytics with Privacy-aware MLOps.
DOI: 10.5220/0011113300003182
In *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022) - Volume 2*, pages 213-220
ISBN: 978-989-758-562-3; ISSN: 2184-5026

presents machine learning operations (MLOps) for education and learning analytic systems, started by the introduction of MLOps approaches for analytics. The next chapter merges MLOps and learning analytics and reviews how learning analytics can be used to enhance the LMSs. After that, we present a few approaches to privacy-aware machine learning, necessary to keep students' data safe and secure. Finally, we illustrate the data flow from the current LMSs to LAOps, which is the term coined to describe learning-analytics-targeted MLOps. Chapter 6 summarises the anticipated benefits and disadvantages for both students and teachers.

2 MLOps FOR EDUCATION AND LEARNING ANALYTIC SYSTEMS

In the digitizing world, enormous amounts of data are collected into various databases. Machine learning (ML) utilises data to provide predictions and recommendations. Especially, the steps taken in the field of deep neural networks to improve prediction accuracy combined with improved computational capacities have enabled its use in complex data sets and increased interest in ML. This increased interest in applying ML to analytics has created demands for easily accessible ML pipelines. In addition, there have been increasing amounts of requests for more sensible reuse of ML components and data cleaning code (Fursin, 2020).

The steps made in the field of DevOps to ease and automatise software development has motivated to implement similar approaches to the field of ML. MLOps is an approach to make the ML pipeline collaborative, reproducible, reusable, and trustable.

Figure 1 displays an overview of the MLOps pipeline, which is simplified from one of the most famous versions of MLOps, Continuous Delivery for Machine Learning (CD4ML) (Sato et al., 2019), and the other MLOps version (Granlund et al., 2021). It can be seen from Figure 1 that the MLOps pipeline can be divided into three parts: 1) the data science part, where the models are built, experimented and evaluated; 2) the actual ML model part; and 3) the production part, where the model is deployed, taken into production, and monitored.

In the data science part of the MLOps pipeline, data engineering is often a considerable task due to various data formats and computer systems. The quality of the curated learning data is paramount for accurate prediction (Renggli et al., 2021; Mäkinen et al., 2021; Valohai ltd., 2020). Data collection must be planned right from the start, preferably with data scientists, to ensure the quality of the data and that the data is easily accessible from relevant databases. Data quality metrics can be used to indicate how quality criteria correlate with the learning process, and how this should reflect in the MLOps pipeline design (Renggli et al., 2021).

Monitoring is an important part of the MLOps to ensure the performance of the system and to restart the model building if the performance starts to deviate. The cycle to restart the model building and update the model depends on the application. Developing predictive ML models may be laborious, the access to the training data may be limited, or there are reasons that the model is wanted to be frozen, such as in some critical medical applications (Granlund et al., 2021). Thus, the motivation to start building a new ML model is higher than it would be in some other application.

MLOps is especially suited for environments, where ML is used either in multiple organisations or in hybrid-cloud environments (Granlund et al., 2021; Banerjee et al., 2020) since the idea of the MLOps is that all the parts of the system should efficiently communicate with each other, there is version control so that the results are reproducible, and the tools should be reusable. An increasing number of companies provide services at least to some part of the MLOps pipeline. For instance, Google Cloud AutoML advertises itself to enable "high-quality custom machine learning models with minimal effort and machine learning expertise."

The explainability and sustainability of models is on the rise, and they are also mandated by the EU regulations in the case of critical applications (Tamburri, 2020). A well-known challenge of especially deep neural networks based ML algorithms is that the explainability of the output is limited so that the algorithms are used more or less as a black box (Adadi and Berrada, 2018). There are research efforts to increase the explainability of the deep neural networks, such as heatmap analysis developed for the image data sets (Borg et al., 2021). These kinds of analysis blocks should be installed in the MLOps pipeline to test the bias of the model and to increase the transparency of the system. The other challenge of the ML systems is that the training data may be biased by gender, ethnically or in some other way causing ethical issues (Safdar et al., 2020). The bias can be alleviated e.g. by slicing the training data set in different ways.

In the proposed system, the ML is meant to give recommendations and to improve students' and teaching personnel's situational awareness. The course

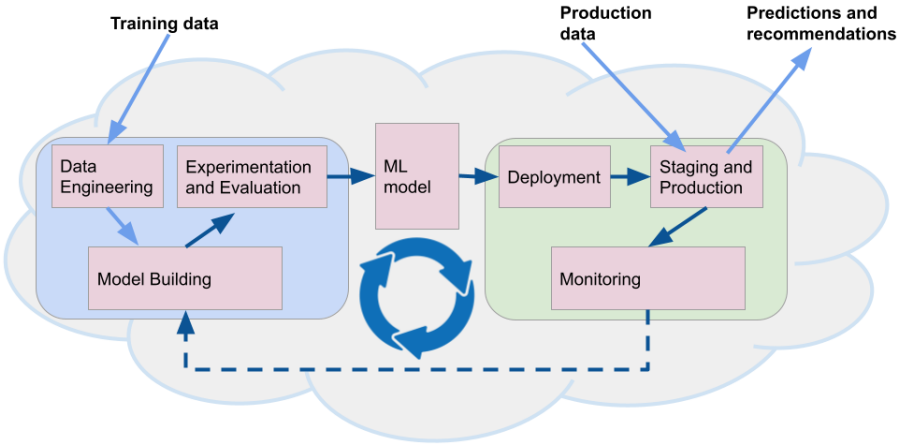

Figure 1: An overview of the MLOps pipeline.

grading is done separately from the ML; hence, the possibly biased training data does not affect the grading. However, the bias should be kept in mind when developing the ML system so that the system does not amplify the existing structures, e.g. the better students would get better advice. To circumvent the challenge, students could see all the possible advice, while the system recommends the advice that is expected to be the most helpful. The students mark their preference, thereby teaching the system for more accurate advising in the future.

3 LMS ENHANCED BY LEARNING ANALYTICS

As group sizes increase, it is no longer possible for course personnel to know their students or establish any more meaningful relationship with them. In this situation, the importance of a pedagogically functioning LMS and good communication practices becomes central. The pedagogy-savvy LMS may take the form of an interactive eTextbook, provide so-called ”smart content” and include features of intelligent tutoring with personalized exercises. Caricatured, this kind of a tutor takes responsibilities that used to belong to teachers.

In Finland, examples of more interactive LMSs are the following: TIM (short for The Interactive Material), VILLE and Plussa. TIM is developed by the University of Jyväskylä and it realizes an eTextbook-type of approach, which besides interactive content enables, e.g., students to leave notes to the side of lecture slides, making it easy for the instructor to pick out the objects to improve (Isomöttönen et al., 2019; Tirronen et al., 2020). Of the mentioned LMSs, VILLE excels in learning analytics (Rajala et al., 2007; Laakso et al., 2018). VILLE group has systematically developed the environment since 2007, spearheaded by analysis and research purposes. Currently in VILLE, formative assessment can be used as an early indicator of dropping out and as a prompt for course personnel to start necessary countermeasures (Veerasamy et al., 2021)

Plussa enables such smart content as programming content examples, programming instructions (Hosseini et al., 2020; Brusilovsky et al., 2018), and animations of algorithms (Sirkiä, 2018; Hosseini et al., 2020). In Plussa, exercises can even be fully-fledged DevOps-simulating assignments where student groups follow the agile methodology, and graders execute unit and end-to-end tests after each submission (Nurminen et al., 2021). One of the best features of Plussa is its microservice-based architecture which makes it easily extendable, exemplified by the easy introduction of new graders (Niemelä and Hyyrö, 2019). Another is the support for Learning Tools Interoperability (LTI) protocol (IMS Global Learning Consortium and others, 2019) that would enable plugging-in exercises in external servers (Manzoor et al., 2019), even if they resided in different continents (Brusilovsky et al., 2018).

Machine learning adds value to the LMS by collecting statistics, by automatic elaborations of the course content (both learning material and material produced by students), and by tracing and modelling the behaviour of students, which in turn enables differentiation, personalising and customisation of exercises. Automatic elaboration would mean, e.g., extractions of concepts and keywords, and automatic annotations. To be able to identify the key concepts is crucial in comprehending topics to learn. As a subject, computer science is similar to mathematics that is built on learning trajectories where conceptual understanding gradually develops and deepens, while

the concepts gradually become more complex. The concepts must be internalized which implies linking the concept to adjacent and parallel concepts in the schema. If the concept is crucial for progress, i.e., only partial or no comprehension will deteriorate or prevent the progress, the concept is called a threshold concept (Boustedt et al., 2007). Automatic detection of such concepts would be helpful.

Tracing the student behaviour may also lead to material rearrangements, if it is noticed that the learning trace is unnecessarily complicated by students having to bounce from one place to another in the material. The learning process information can also be processed as recommendations, such as suggestions for extra reading, links to helpful exercises, in particular such exercises that would better help to solve the troublesome task at hand. For example, if it is found that a similarly profiled student appears to be particularly benefiting from a particular representation or receiving a eureka moment after some programming content examples, the very same material may be offered to another student in the corresponding profile category. Irrespective of ML usage, the recommendation and hints can also be queried directly from students as one type of exercise and offered to other students of the same profile in the same cluster, in a crowd-sourced manner. Within clusters, students share some kind of submissions, activity, and performance level. The clusters can also be utilized in group formation, to achieve more reasoned divisions.

LMS can be built to support the weak and to engage ill-motivated students. To avoid frustration/boredom, the challenge level must be in accordance with the skill level of a student. Preferably, LMS dynamically differentiates the challenge level by having collected information on students' performance. In automatized follow-ups, LMS can also utilize students' self-reflections and such commitments as a target grade in obligating changes to activity level, if needed. Reflective discussions could be handled by a chatbot, e.g., in situations where results deteriorate rapidly. With an analogue of a sports watch, LMS can serve students with statistics, and increase their awareness of one's own performance in relation both to previous personal performance and of the one of the whole group. Comparably to sports watches, formative assessment on-a-go is anticipated to improve self-efficacy and self-regulation, needed in distance learning and MOOCs, where learning is more autonomous by definition.

Learning analytic mandates modelling of a student. The modelling has developed considerably in recent years and there are several different alternatives available, e.g., student modelling (Chau et al., 2021), open student modelling (Bull and Kay, 2013), even open social student modelling (Brusilovsky and Rus, 2019). Modelling is such a big effort, so it would be desirable for the models to be interchangeable between different courses, even different systems. Reuse would require smarter mapping, e.g., with the help of ontologies (Chau et al., 2021).

Naturally, in the beginning, the setup of an ML system requires extra time and effort. After drivenin, the disadvantages of ML usage are increased control and decrease of privacy, at least if privacy issues are dealt poorly with the LMS. The potential opacity of the system would stir up fears of misuse of data. Preferably, a teacher could consult an expert, address the ethical concerns in particular, and ensure transparency as far as possible. Transparency is increased also by informing students about on-going analyses and research aims. It is good to explain what is the rationale behind automatically made conclusions, i.e., to respond to the demand for explainability of background algorithms. In compliance, a few articles emphasized the need for more controllable and explainable recommendations to add to the transparency of the system (Chacon and Sosnovsky, 2021). Also, if data must be transferred elsewhere to enable heavier computations, it must be handled with all the needed care and safety.

4 PRIVACY-AWARE MACHINE LEARNING

In many cases, LMSs store user data about both learners and educators. Such data can include personal details, as well as interactions, navigation patterns, learning patterns, results and grades obtained, communication with other actors as well as possible reasons where anomalous behaviour (such as long periods of absence, or unusual and different quality of submissions) is noticed. While many LMSs are running on each organisation's own infrastructure, in many instances, such data is stored or sent to the cloud, where machine-learning algorithms and applications may be employed to derive subsequent metadata. Such a cloud-based service may be offered by an external or commercial service provider. A major challenge here is the protection and privacy of such sensitive user data when considering cloud-based compute and storage services. Sensitive and private data may unintentionally or deliberately be made available to unauthorised third parties or insiders, including situations where these LMSs may even run on local infrastructure. Nevertheless, the overarching issue remains unchanged: Preventing access to

data stored in any locations to any user without the correct credentials. In addition, the different means of protecting data needs to mirror the wide variety of stakeholders and organisations.

Today, data is usually sent to a cloud service provider (CSP) using a RESTful API over a secure channel. This typically implies transport layer security, in which a server is validated by a certificate. By default, a user sends data without encryption. However, the CSP must ensure that data is stored and maintained in an encrypted form. Most existing approaches do not manage to protect users' data against attacks of either adept external adversaries or insiders. Two properties of the CSP can be attributed to this inadequacy: a bulk data storage and encryption function. Vulnerabilities in these CSP's properties attract attackers. An attacker could even become privy to the data, having mounted a successful attack. Research in the field of Symmetric Searchable Encryption (SSE) provides promising solutions to overcome such challenges.

SSE schemes allow users to encrypt data with a symmetric key unknown to the CSP and search directly over the encrypted data. However, SSE schemes discourage a user from sharing data with other users, as sharing a symmetrically-encrypted file requires sharing the secret key. As a result, when a user needs to be revoked, the data owner needs to re-encrypt the data with a fresh key and distribute the new key to the remaining legitimate users. To address the problem of revocation, researchers proposed solutions based on Attribute-Based Encryption schemes (ABE) (Michalas, 2019).

ABE schemes allow users to encrypt a file based on a certain policy. Then, a unique key is generated for each user that has access to the CSP resources. This key is generated based on a list of attributes. A user can decrypt a file encrypted with a certain policy if and only if the attributes of her key satisfy the underlying policy. In this case, access revocation is more efficient: it only requires revoking the corresponding user's key, while the keys of all other users remain the same. However, the access control flexibility enabled by ABE schemes comes at a cost: the generated ABE cipher-texts are rather large. As a result, decryption requires significant computational resources (Michalas, 2019).

Another challenge in LMSs is to run statistical and analytical ML functions on large datasets without revealing the identities of individuals related to the data. This challenge can be addressed by the design and implementation of a set of protocols based on another promising cryptographic primitive called Functional Encryption (FE). Functional encryption is a new paradigm in public-key encryption that allows users to finely control the amount of information that is revealed by a ciphertext to a given receiver (Abdalla et al., 2015). Thus, statistical computations can be performed based on encrypted data allowing sensitive data to remain protected during computation. Analysts will be unable to obtain revealing data as a consequence.

To protect the privacy of users and prevent several types of malicious behaviour (which includes insider threats), several types of modern cryptography approaches are needed to construct a privacy-aware cloud-based LMS. Privacy protection can be strengthened in different ways, e.g.: by enabling SSE and ABE to use both encryption schemes in the most efficient way in order to store and process user data, or by utilizing effective user access revocation approaches that do not affect other users or the overall functionality of the service. These enable education professionals to generate analytics and statistical measurements in a privacy-preserving way.

5 LAOps DATAFLOW

Currently deployed teaching systems at Tampere University include LMSs like Moodle, Plussa, and Gitlab; others exist yet lesser in relevance. During programming courses, students commit their code to repositories provided to them using Gitlab. The data from LMSs and Gitlab is the main focus of the proposed LAOps solution. Figure 2 displays an example of how data from current LMS systems' databases could flow to secure LAOps processes in the cloud. In the image, we can see two possible approaches for fetching data to LAOps, which can be applied simultaneously. In the first approach, LAOps processes periodically fetch data through the APIs provided by the LMS components. In the latter approach, LMS components that have Continuous Integration or Continuous Delivery pipelines integrate sending data to LAOps processes into these pipelines.

The steps listed below detail the process for the fore-mentioned two approaches:

1. Instructor constructs upstream repositories: student_template_project and group_template_project with the help of a self-made Gitlab management tool; these upstream repositories are updated regularly by the instructor.
2. Student pulls upstream, gets instructions, a) commits his/her code either to student repository if submitting alone, or b) to group repository in case of group work.

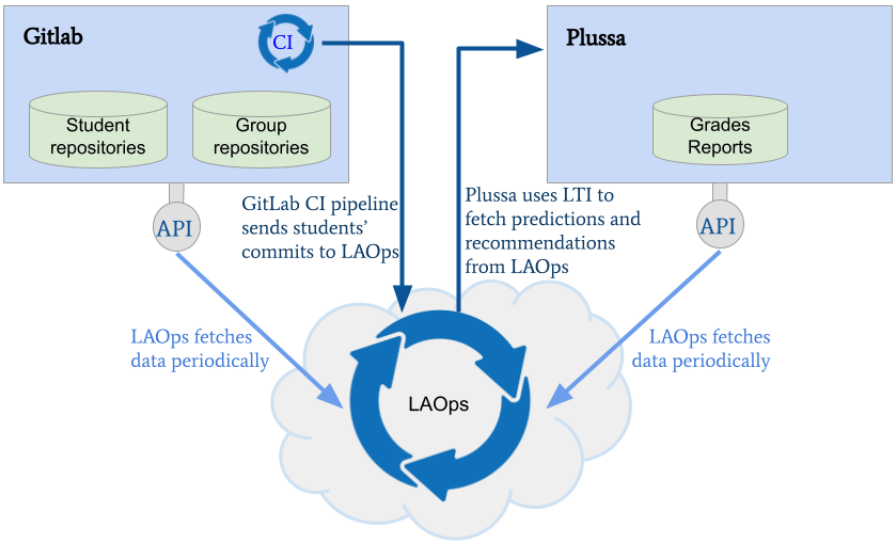

Figure 2: The interplay of Plussa, GitLab and secure LAOps in the proposed scenario.

3. After content with the code, a student initiates the grading in Plussa learning management system by submitting their git URL there.
4. Plussa clones and grades the student's submission, and the generated grades and reports are stored in Plussa's database.
5. A student's commit to Gitlab launches a Gitlab CI pipeline, which has been set up to send the anonymised data to LAOps.
6. Plussa provides a RESTful API (no CI/CD pipeline), LAOps components will periodically fetch anonymised data through its API.
7. LAOps performs MLOps as presented in Figure 1: LAOps saves the incoming data to the database in a privacy-secure way, gives predictions and recommendations based on the developed ML algorithms, and monitors the performance of the algorithms to initialize the updates to the algorithms.
8. The necessary scripts for handling and visualising the dataset that LAOps accumulates are available in upstream repositories. Students may, if interested, check the statistics and see the dashboard of the progress of the course.

The anonymisation of the student data is crucial, as it enables using the data in analytics performed in the cloud resources located outside the control of Tampere University. However, most LMSs have APIs which give students data as-is, uncensored. This necessitates that anonymisation is implemented now as part of this proposal.

During the risk analysis stage other data security matters pertaining to using sensitive personal data with LAOps were identified. These included the privacy matters as discussed in the previous chapter, as well as other general data security concerns including:

- which access control mechanisms and processes need to be put in place in the cloud as well in our research group so that access to the data and sections of data can be limited on per user basis
- how to find and apply CSP-dependant best practices on building cloud systems in a way that reduces or even eliminates the chance of data breaches
- how we can demonstrate that the process we use for anonymisation of the personal data is on sufficient level, and able to cope with changes in the learning systems from where data is fetched.

6 CONCLUSIONS

We have proposed an architecture for learning analytic system, LAOps, which is secure and adaptable for both students and teachers. The architecture is motivated by the recent developments of MLOps as well as privacy-aware cryptographic data storage in the cloud. The aimed benefits of LAOps for the student:

- better-informed intelligent tutoring, scaffolding
- personalisation / customisation / differentiation
- recommendations (reading recommendations, useful assignments), analogies, metaphors
- statistics, awareness of one's own performance in relation to the whole group

- increased self-direction and autonomy
- better-justified creation of groups based on profiling.

Benefits for the teacher:

- better statistics / better understanding of the current status
- profiling of students, e.g., for early detection of dropouts, or teamwork problems
- detection of useful vs. challenging tasks, more precise identification of threshold concepts
- hints for material improvement, rearrangement and replenishment
- automatic annotations, keyword search.

The usage of cryptographic schemes for data storage as well as for privacy-preserving analytics, is essential to alleviate fears of leakage of private data to untrusted third parties. Additionally increased user involvement and awareness that the LMS has been designed with security in mind, aid in reducing perceptions that invasive machine learning methods are employed on personally identifiable information. The regulation of LA is under development, new stricter and refined legislation and rules are to be expected.

REFERENCES

Abdalla, M., Bourse, F., Caro, A. D., and Pointcheval, D. (2015). Simple functional encryption schemes for inner products. In *IACR International Workshop on Public Key Cryptography*, pages 733–751. Springer.

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160.

Banerjee, A., Chen, C., Hung, C., Huang, X., Wang, Y., and Chevesaran, R. (2020). Challenges and experiences with MLOps for performance diagnostics in hybrid-cloud enterprise software deployments. In Talagala, N. and Young, J., editors, *2020 USENIX Conference on Operational Machine Learning, OpML 2020, July 28 - August 7, 2020*, pages 37–39. USENIX Association.

Borg, M., Jabangwe, R., Åberg, S., Ekblom, A., Hedlund, L., and Lidfeldt, A. (2021). Test automation with grad-CAM Heatmaps - A future pipe segment in MLOps for Vision AI? In *14th IEEE International Conference on Software Testing, Verification and Validation Workshops, ICST Workshops 2021, Porto de Galinhas, Brazil, April 12-16, 2021*, pages 175–181. IEEE.

Boustedt, J., Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., and Zander, C. (2007). Threshold concepts in computer science: do they exist and are they useful? *ACM Sigcse Bulletin*, 39(1):504–508.

Brusilovsky, P., Malmi, L., Hosseini, R., Guerra, J., Sirkiä, T., and Pollari-Malmi, K. (2018). An integrated practice system for learning programming in Python: design and evaluation. *Research and Practice in Technology Enhanced Learning*, 13(1):1–40.

Brusilovsky, P. and Rus, V. (2019). Social navigation for self-improving intelligent educational systems. *Design Recommendations for Intelligent Tutoring Systems*, page 131.

Bull, S. and Kay, J. (2013). Open learner models as drivers for metacognitive processes. In *International Handbook of Metacognition and Learning Technologies*, pages 349–365. Springer.

Chacon, I. A. and Sosnovsky, S. A. (2021). Knowledge models from PDF textbooks. *New Review of Hypermedia and Multimedia*, 27(1-2):128–176.

Chau, H., Labutov, I., Thaker, K., He, D., and Brusilovsky, P. (2021). Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*, 31(4):820–846.

Fursin, G. (2020). The Collective Knowledge project: making ML models more portable and reproducible with open APIs, reusable best practices and MLOps. *CoRR*, abs/2006.07161.

Granlund, T., Kopponen, A., Stirbu, V., Myllyaho, L., and Mikkonen, T. (2021). Mlops challenges in multi-organization setup: Experiences from two real-world cases. In *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, pages 82–88. IEEE.

Heikkilä, H. and Okkonen, J. (2021). AI driven competency development at the threshold of working life. In *2021 Nordic Learning Analytics (Summer) Institute, NLASI 2021*. CEUR-WS.

Hosseini, R., Akhuseyinoglu, K., Brusilovsky, P., Malmi, L., Pollari-Malmi, K., Schunn, C., and Sirkiä, T. (2020). Improving engagement in program construction examples for learning Python programming. *International Journal of Artificial Intelligence in Education*, 30(2):299–336.

IMS Global Learning Consortium and others (2019). Learning tools interoperability core specification. *IMS Final Release Version*, 1.

Isomöttönen, V., Lakanen, A.-J., and Lappalainen, V. (2019). Less is more! Preliminary evaluation of multi-functional document-based online learning environment. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE.

Laakso, M.-J., Kurvinen, E., Enges-Pyykönen, P., and Kaila, E. (2018). Designing and creating a framework for learning analytics in Finland. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0695–0700. IEEE.

Mäkinen, S., Skogström, H., Laaksonen, E., and Mikkonen, T. (2021). Who needs MLOps: What data scientists seek to accomplish and how can MLOps help? In *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, pages 109–112. IEEE.

Manzoor, H., Akhuseyinoglu, K., Shaffer, C., and Brusilovsky, P. (2019). OpenDSA/Mastery Grids Exercise Interchange. In *Proceedings of the Fourth SPLICE Workshop colocated with 50th ACM Technical Symposium on Computer Science Education (SIGSCE 2019), Minneapolis, MN, USA*.

Michalas, A. (2019). The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 146–155.

Niemelä, P. and Hyyrö, H. (2019). Migrating learning management systems towards microservice architecture. *Joint Proceedings of the Inforte Summer School on Software Maintenance and Evolution (SSSME-2019)*, page 11.

Nurminen, M., Niemelä, P., and Järvinen, H.-M. (2021). Having it all: auto-graders reduce workload yet increase the quantity and quality of feedback. In *SEFI Annual Conference: Blended Learning in Engineering Education: challenging, enlightening–and lasting*, pages 385–393.

Okkonen, J., Helle, T., and Lindsten, H. (2020a). Ethical considerations on using learning analytics in Finnish higher education. In *International Conference on Applied Human Factors and Ergonomics*, pages 77–85. Springer.

Okkonen, J., Helle, T., and Lindsten, H. (2020b). Expectation differences between students and staff of using learning analytics in Finnish universities. In *International Conference on Information Technology & Systems*, pages 383–393. Springer.

Rajala, T., Laakso, M.-J., Kaila, E., and Salakoski, T. (2007). VILLE: a language-independent program visualization tool. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88*, pages 151–159.

Renggli, C., Rimanic, L., Merve Gürel, N., Karlaš, B., Wu, W., and Zhang, C. (2021). A data quality-driven view of MLOps. *arXiv e-prints*, pages arXiv–2102.

Safdar, N. M., Banja, J. D., and Meltzer, C. C. (2020). Ethical considerations in artificial intelligence. *European Journal of Radiology*, 122:108768.

Sato, D., Wilder, A., and Windheuser, C. (2019). Continuous delivery for machine learning. https://martinfowler.com/articles/cd4ml.html accessed Feb 15, 2022.

Sirkiä, T. (2018). JSVEE & Kelmu: Creating and tailoring program animations for computing education. *Journal of Software: Evolution and Process*, 30(2):e1924.

Tamburri, D. A. (2020). Sustainable MLOps: Trends and challenges. In *22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2020, Timisoara, Romania, September 1-4, 2020*, pages 17–23. IEEE.

Tirronen, V., Lappalainen, V., Isomöttönen, V., Lakanen, A.-J., Taipalus, T., Nieminen, P., and Ogbechie, A. (2020). Incorporating teacher-student dialogue into digital course material: Usage patterns and first experiences. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE.

Valohai ltd. (2020). Practical MLOps.

Veerasamy, A. K., Laakso, M.-J., and D'Souza, D. (2021). Formative assessment tasks as indicators of student engagement for predicting at-risk students in programming courses. *Informatics in Education*.