P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

Adaptive AI Pipelines for Smart Operations

Sophia Martinez
Asst Professor

Abstract: Artificial intelligence (AI) is being deployed at an accelerating pace across production systems, yet the observability and adaptability of AI service delivery remain underexplored. This work examines the need for real-time operational intelligence within AI pipelines and proposes architectural principles for adaptive, production-grade AI service delivery. Performance considerations for time-sensitive applications are addressed through dynamic pipeline architectures, incorporating early-exit strategies, adaptable inference models, and resource-aware approaches to autoscaling and scheduling. These concepts are evaluated using an AI pipeline for classical music genre classification, where pipeline quality is governed by delivery-performance metrics benchmarked against ground-truth data. Real-time observability components are traced within a real-world AI service to identify performance bottlenecks and inform adaptive optimization strategies.

Keywords :Performance-Aware Artificial Intelligence,AI Pipelines,Adaptive Service Delivery,Operational Intelligence,Intelligent Workflow Optimization,Real-Time Analytics,Predictive Decision Support,Machine Learning Operations (MLOps),Scalable AI Systems,Data-Driven Service Optimization.

1. Introduction

Modern applications increasingly depend on artificial intelligence (AI) and machine learning (ML) to enhance their functions and create advanced services. These deep integrations of AI/ML introduce a new layer of complexity for both the service delivery and operational management of applications compared with traditional approaches. Combinations of multiple AI/ML services into AI pipelines further amplify the complexity, particularly when such pipelines execute continually and adaptively in response to variant input data streams, perhaps even supported by continuous retraining of pipelines, components, and/or scaling. Modern application architectures must thus accommodate not only the expected sources of performance degradation such as classic performance bottlenecks but also a new set of AI-related performance issues, including data and service fluctuations and drift, model staleness, and dynamic load changes.

Total Pipeline Latency

$$L_{total} = \sum_{i=1}^{n} L_i$$

Where $L_i$is the latency of each pipeline stage.

Current approaches to performance management often focus on only a subset of these issues and lack the granularity required to target core AI/ML aspects. AI pipeline development and operation require dedicated expertise that is not always present. To overcome these limitations, a dedicated set of performance-aware architectural principles is proposed. These principles establish a foundation for enhancing service delivery through adaptive pipelines and for operational management through business-centric, application-wide operational intelligence. Together, these capabilities enable the management of complex AI pipelines and their dynamic component services simply and effectively.

P-ISSN: 3050-9572
E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

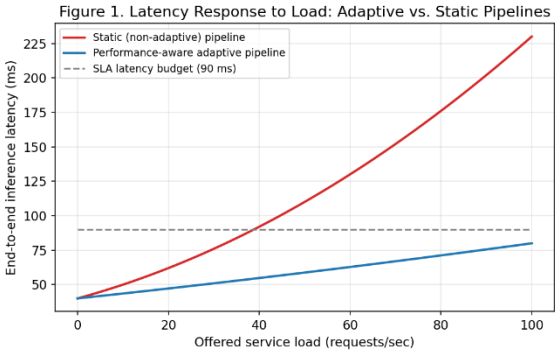


Fig 1:

A. Architectural principles for operational intelligence

Large-scale machine learning (ML) systems deployed in production need to guarantee a quality of service that meets user expectations. These expectations evolve over time based on changing circumstances and requirements, and can be expressed in terms of performance metrics such as service time, service quality, and service availability. However, users also expect these system to be able to deliver adaptive services: under-utilized resources should be released, and low-quality service pragmatically compensated for by improved performance or reduced costs. Enabling these capabilities for AI service delivery requires a dedicated design and construction effort focused on software systems specifically built to deliver performance-aware machine-learning services.

Service Quality Score

$$Q_s = \alpha A + \beta P - \gamma L$$

Where $A$is accuracy, $P$is availability, and $L$is latency.

The feedback control and tuning of adaptive quality-of-service metrics requires software instrumentation to provide real-time feedback loops and system autoregulation in relation to those attributes, together with resource-aware scheduling and autoscaling capabilities across the entire service delivery stack. Observability, in terms of runtime monitoring and tracing, is key to expressing any quality-of-service attribute in terms of application-specific metrics (and their associated monitoring instrumentation) as well as being able to diagnose the reasons for (and automatically correct) delivery deviations. Such operational-intelligence capabilities can be enabled using concepts and principles inherited from Site Reliability Engineering (SRE) and platform operational maturity levels.

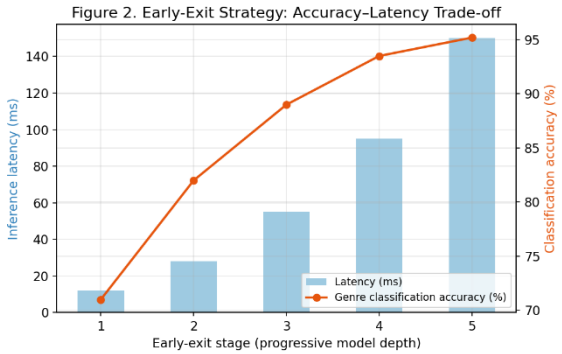


Fig 3:

2. Foundations of Performance-Aware AI Pipelines

Adaptive service delivery and operational intelligence require responsive and resilient AI pipelines that support real-time and near real-time machine learning tasks. Metric-driven pipelines with feedback loops that support adaptive service delivery are a key requirement. Real-time monitoring and alerting, together with resource-aware scheduling and autoscaling rules for declarative AI-service deployment, can provide operational resilience for AI pipelines. An operational intelligence framework for Machine Learning Operations (MLOps) and AI pipelines addresses the challenges of observability and performance monitoring in

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

production AI models, including detecting anomalous behavior, analyzing associated events across AI pipelines, and debugging common errors in AI models. The proposed framework provides a mechanism to add observability-related traces and a layer of debugging checks (for example, schema validation) to complex AI pipelines through simple configuration.

Adaptive Resource Allocation

$$R_t = R_{t-1} + \Delta W_t$$

Where $R_t$is current resource allocation and $\Delta W_t$is workload change.

The service demand for many applications has highly fluctuating patterns in nature, and even may consist of unexpected extreme peaks that last for a long duration. In these scenarios, it is very difficult to accurately configure service resources for optimal operational performance and cost-effectiveness. Real-time detection of sudden service pattern changes and automatic adjustment of service resources for response can solve this problem. In a performance-demand adaptive service delivery model, such AI service demand fluctuations can be monitored by an augmented multi-dimensional adaptive monitoring service. Performance-demand-sensitive metrics and/or performance-demand-sensitivity of the metrics are generated in real time for those pipelines that have dynamic models loaded inside, and these metrics are used to steer real-time resource management of the service-hosted resources.

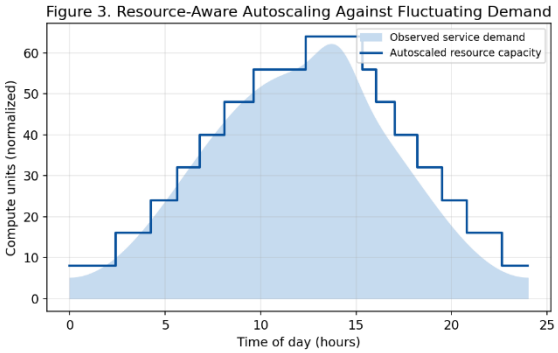


Fig 3:

A. Metrics and benchmarks for adaptive service delivery

Designing a decision-support service requires consideration of the data and models to be used, together with the expected load on the service. For many types of AI services, however, the data, models and their load vary considerably over short time scales, necessitating adaptation in the service delivery approach. Key indicators that need to be monitored in such cases relate to the expected latency and quality of the prediction. Unlike standard service response times, during higher than expected load these metrics can actually exceed agreed service-level guarantees. Hence the monitored indicators do not directly map to concrete low-level resource constraints, but are at a higher level that represents a form of adaptive filtering. Since constant monitoring introduces overhead, it should only be engaged for critical or early-warning situations.

Autoscaling Decision

$$S = \begin{cases} Scale\ Up, & W_t > T_h \\ Scale\ Down, & W_t < T_l \end{cases}$$

Where $W_t$is workload, $T_h$is high threshold, and $T_l$is low threshold.

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

For decision-support services built around critical error-checking and analysis, progressive verification can also be used in conjunction with adaptive load thresholds. Here models can be applied sequentially, with those of lower quality used as early warning indicators, followed possibly by more accurate but slower models if time permits. The AI service thus consists of a dynamically constructed pipeline of stages, each of which may itself be a distinct AI service.

3. Adaptive Service Delivery through Dynamic Pipelines

The dynamic adaptation of pipelines during service delivery provides apparent advantages when training AI models over long periods of time and it can also be exploited by operational intelligence. For example, the configuration of data collection and preprocessing components can depend on user-provided information, while parts of the pipeline dedicated to model serving can change automatically depending on resource availability. While this dynamic adaptation can certainly leverage the entire pipeline, it is a rather narrow view of dynamic pipelines. The levels of the serveless architecture for AI-based services encompass all AI-related activities for an organisation and pipelines may be dynamically adapted whenever needed for the service delivery, even when the main components of the workflow are not explicitly fed by the service request or response.

Model Selection Rule

$$M^* = \arg\min_M (L_M + C_M)$$

Where L_M is model latency and C_M is model cost.

Dynamic adaptation of pipelines during service delivery has been pleasantly demonstrated through example. Real-time monitoring and feedback loops are generalised concepts crucial for the success of any large-scale IT operation, and resource-aware scheduling and autoscaling have direct implications for business key performance indicators. In the context of AI services, these ideas are often connected to the concept of operational intelligence and addressed through system-level metrics, response-time budgets, and high-level service-level objectives.

Table 1. Core Components of Performance-Aware AI Pipelines

Component	Purpose	Expected Benefit
Real-Time Monitoring	Tracks pipeline behavior continuously	Faster issue detection
Feedback Loops	Adjusts pipeline operation dynamically	Adaptive service quality
Resource-Aware Scheduling	Allocates resources based on workload	Reduced latency and cost
Autoscaling	Expands or reduces resources automatically	Handles demand fluctuations
Observability Layer	Provides logs, traces, and metrics	Better debugging and reliability

A. Real-time monitoring and feedback loops

Real-time monitoring of ML pipelines enables closed-loop feedback in adaptive service deployment. Feedback loops depend on different types of real-time metrics: source metrics, application metrics, and online learning feedback.

Machine learning and AI systems generate real-time monitoring data of significant volume and variety, from

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

sources not traditionally monitored in IT. As well as well-defined production metrics, there are many relationship-based metrics that can provide insights into resource usage. Upstream monitoring of data inputs, data quality, and distribution shifts is also key to ensuring continual service quality.

Prediction Quality

$$Q_p = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP, TN, FP, FN represent classification outcomes.

Real-time feedback on ML and AI pipeline data sources and application environments allows operational monitoring to respond quickly and automatically to unplanned demand or trends. Dynamic capacity and performance management respond to varying resource loads or platform performance levels, auto-scaling or augmenting resources based on detected performance shortfalls. Dynamically driven changes to AI and ML models, algorithms, and pipelines adapt service delivery for continual quality improvement.

Table 2. Key Performance Metrics for Adaptive Service Delivery

Metric	Description	Relevance
Latency	Time taken to return prediction/service output	Measures responsiveness
Service Quality	Accuracy or usefulness of AI output	Ensures reliable decisions
Availability	Service uptime and accessibility	Supports dependable delivery
Resource Usage	CPU, memory, GPU, or cluster utilization	Optimizes operational cost
Error Rate	Frequency of failed or incorrect outputs	Indicates service stability

B. Resource-aware scheduling and autoscaling

Efficient resource scheduling is essential to minimize the cost of serving tasks while meeting latency objectives. AI pipelines usually combine long-latency components (training or high-throughput ones) with low-latency ones. Such temporal imbalances are frequently exploited to reactively scale the resources allocated to low-latency components. A resource-aware scheduling can furthermore achieve a significant cost-reduction by proactively stopping low-latency components when the offered load is small and only rewarming them when the estimated workload is high.

Monitoring Overhead

$$O_m = \frac{T_m}{T_p}$$

Where T_m is monitoring time and T_p is total pipeline processing time.

Flexible autoscaling mechanisms ensure that the allocated resource adapts to workload fluctuations. Changing the size of clusters with short-lived or bursty workloads or when the maximum number of tasks is unpredictable is

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

complex. Nevertheless, during such momentary spikes, the processing of the intermediate files may remain a bottleneck. Kubernetes scaling can be applied both vertically-and-horizontally. Within such complex settings, it allows rapid cluster expansion without adding the cloud more containers and taking advantage of the fact that nodes sometimes remain idle.

Table 3. Common Challenges in AI Pipeline Operations

Challenge	Cause	Possible Solution
Model Staleness	Outdated trained models	Continuous retraining
Data Drift	Changing input patterns	Data distribution monitoring
Load Spikes	Sudden traffic increase	Autoscaling
Pipeline Bottlenecks	Slow intermediate stages	End-to-end tracing
Manual Debugging	Lack of visibility	Automated observability tools

4. Operational Intelligence for AI Pipelines

AI pipelines are becoming indispensable for organizations, yet their internal workings remain largely opaque. As a result, incident detection and root-cause analysis need to be performed manually, undermining their reliability and robustness. Addressing these issues requires observability, tracing, and debugging. While these ideas are established in the context of microservices and distributed systems, existing AI pipelines frameworks fail to provide the necessary evidence to take advantage of them. To overcome this limitation, recent work proposes an augmenting architectural approach that provides all the essential components to implement operational intelligence around AI pipelines.

Drift Detection Score

$$D = \mid P_{train}(x) - P_{live}(x) \mid$$

Where P_{train} and P_{live} represent training and live data distributions.

Even with this information, AI pipelines require a broader class of observability, support for tracing end-to-end data flows from sources to sinks, and tools for improper-class debugging. Operators must verify whether the outcome provided by the AI system, together with its internal states, is consistent and expectable concerning the whole, complex workflow. End-to-end tracing makes it possible to track data as they flow across multiple interconnected resources and components while monitoring the runtime to detect anomalies in the results.

Table 4. Monitoring Types in AI Pipelines

Monitoring Type	Focus Area	Example Use
Source Monitoring	Input data and quality	Detecting data drift
Application Monitoring	Service-level behavior	Tracking latency
Model Monitoring	Prediction quality	Identifying accuracy drops
Resource Monitoring	Infrastructure usage	Scaling CPU/GPU resources

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

Monitoring Type	Focus Area	Example Use
Trace Monitoring	End-to-end request path	Finding slow pipeline stages

A. Observability, tracing, and debugging in complex workflows

Monitoring and debugging represent a paramount challenge for complex AI service workflows. The richness of modern AI pipelines, integrating a multitude of models and components, necessitates dedicated observability mechanisms capable of providing valuable insights during both training and inference phases. Instrumentation for tracing and detailed logging is critical when estimating model responses during inference, yet this is often neglected. A performance degradation during any portion of the pipeline can dramatically impact the overall latency of the workflow, and enrolling an entire pipeline for instrumentation can quickly become untenable.

SLA Violation Rate

$$V_{sla} = \frac{N_v}{N_t}$$

Where $N_v$is the number of violated requests and $N_t$is total requests.

Observing the various components of a pipeline helps to detect failure patterns, while tracing tools enable monitoring of specific branches. During inference, requests can be traced under certain conditions (e.g., user-specified IDs) to provide detailed logging aligned with user services and help pinpoint slows responses in tailored dashboards. Request logs that aggregate errors and warnings also aid in segmenting slow responses across specific parts of the pipeline. These features can be accompanied by automated debugging tools. Previous work describes a solution that combines errors from common component failures with response times on an unmonitored failure edge, enabling automatic detection of possible behaviors behind unexpected model responses. An extended approach reasons about possible component replacements and their impact on anomaly detection. Together, these elements simplify the operational issue of monitoring complex AI workflows.

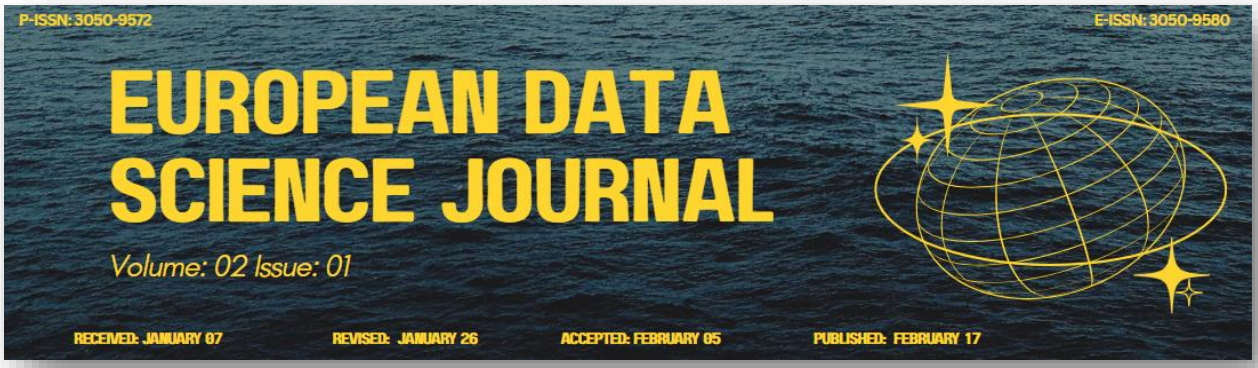
5. Conclusion

The work discussed the requirements and challenges of adaptive service delivery and operational intelligence for AI pipelines. The limitations of conventional monitoring and management solutions become evident as the AI model grows in complexity. Their use may still be appropriate and effective when an AI agent communicates with a simple model running on isolated hardware; however, such approaches cannot provide the responsiveness to meet delay goals or match the service quality of adaptive service delivery.

The conceptual and architectural understandings of Performance-Aware AI Pipelines will help to address these challenges by providing an architectural blueprint for these two closely related aspects. Performance-Aware AI Pipelines are performance-aware, quality-aware, and resource-aware. Their behavior is monitored in real time, and feedback loops enable the automated management of pipelines and the workloads they execute. The pipelines are dynamic in the sense that real-time requirement violations trigger a process to redirect the current workload away from the Performance-Aware AI Pipeline in an effort to meet the delay requirement. During such rerouting, service to altered requests are provided by an alternate model, possibly an inferior one, and running on a less capable resource.

Acknowledgment

This work is based upon research supported by the U.S. Army Research Laboratory under Agreement Number



W911NF-20-2-0007, the U.S. Naval Research Laboratory under Agreements No. N00173-06-2-C019 and No. N00173-20-C-2001, the Office of Naval Research under Grant No. ONR-N00173-19-2-G002, and the National Security Innovation Capital under Agreement Number HQ00342390009B. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the U.S. Army Research Laboratory, the U.S. Naval Research Laboratory, the Office of Naval Research, the National Security Innovation Capital, or the U.S. Government.

References

1. Baier, L., Schlör, T., Schoeffer, J., & Kühl, N. (2023). Detecting concept drift with neural network model uncertainty. In T. X. Bui (Ed.), Proceedings of the 56th Hawaii International Conference on System Sciences (pp. 835–844).
2. Barbudo, R., Ventura, S., & Romero, J. R. (2023). Eight years of AutoML: Categorisation, review and trends. Knowledge and Information Systems, 65, 5097–5149.
3. Chakilam, C., Suura, S. R., Koppolu, H. K. R., & Recharla, M. (2022). From Data to Cure: Leveraging Artificial Intelligence and Big Data Analytics in Accelerating Disease Research and Treatment Development. Journal of Survey in Fisheries Sciences. https://doi. org/10.53555/sfs. v9i3, 3619.
4. Boobalan, P., Ramu, S. P., Pham, Q.-V., Dev, K., Pandya, S., Maddikunta, P. K. R., Gadekallu, T. R., & Huynh-The, T. (2022). Fusion of federated learning and industrial Internet of Things: A survey. Computer Networks, 212, 109048.
5. Esmaeilzadeh, A., Heidari, M., Abdolazimi, R., Hajibabaee, P., & Malekzadeh, M. (2022). Efficient large scale NLP feature engineering with Apache Spark. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC).
6. Granlund, T., Kopponen, A., Stirbu, V. A., Myllyaho, L., & 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).
7. Adusupalli, B. (2023). DevOps-Enabled Tax Intelligence: A Scalable Architecture for Real-Time Compliance in Insurance Advisory. Journal for Reattach Therapy and Development Diversities. Green Publication. https://doi. org/10.53555/jrtdd. v6i10s (2), 358.
8. Hewage, N., & Meedeniya, D. (2022). Machine learning operations: A survey on MLOps tool support. arXiv.
9. Jakubik, J., Vössing, M., Kühl, N., Walk, J., & Satzger, G. (2022). Data-centric artificial intelligence. arXiv.
10. Jan, Z., Ahamed, F., Mayer, W., Patel, N., Grossmann, G., Stumptner, M., & Kuusk, A. (2023). Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. Expert Systems with Applications, 216, 119456.

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

11. Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): Overview, definition, and architecture. IEEE Access, 11, 31866–31879.
12. Lee, C., & Lim, C. (2021). From technological development to social advance: A review of Industry 4.0 through machine learning. Technological Forecasting and Social Change, 167, 120653.
13. Mäkinen, S. S., Skogström, H., Laaksonen, E., & 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).
14. Mashetty, S. (2023). View of A Comparative Analysis of Patented Technologies Supporting Mortgage and Housing Finance. Educational Administration: Theory and Practice.
15. Mikkonen, T., Nurminen, J. K., Raatikainen, M., Fronza, I., Mäkitalo, N., & Männistö, T. (2021). Is machine learning software just software: A maintainability view. In D. Winkler, S. Biffl, D. Mendez, M. Wimmer, & J. Bergsmann (Eds.), Software Quality: Future Perspectives on Software Engineering Quality (pp. 94–105). Springer.
16. Priestley, M., O'Donnell, F., & Simperl, E. (2023). A survey of data quality requirements that matter in ML development pipelines. Journal of Data and Information Quality, 15(2), Article 11.
17. Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and Industry 4.0 applications. International Journal of Production Research, 59(16), 4773–4778.
18. Rajesh, R., et al. (2022). Smart manufacturing through machine learning: A review, perspective, and future directions to the machining industry. Journal of Engineering, 2022, Article 9735862.
19. Renggli, C., Rimanic, L., Gürel, N. M., Karlas, B., Wu, W., & Zhang, C. (2021). A data quality-driven view of MLOps. IEEE Data Engineering Bulletin, 44(1), 11–23.
20. Ruf, P., Madan, M., Reich, C., & Ould-Abdeslam, D. (2021). Demystifying MLOps and presenting a recipe for the selection of open-source tools. Applied Sciences, 11(19), 8861.
21. Paleti, S. (2023). Transforming Money Transfers and Financial Inclusion: The Impact of AI-Powered Risk Mitigation and Deep Learning-Based Fraud Prevention in Cross-Border Transactions. Available at SSRN, 5158588.
22. Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., & Calinescu, A. (2022). Digital twins: State of the art theory and practice, challenges, and open research questions. Journal of Industrial Information Integration, 30, 100383.
23. Zhou, J., Zhang, S., Lu, Q., Dai, W., Chen, M., Liu, X., Pirttikangas, S., Shi, Y., Zhang, W., & Herrera-Viedma, E. (2021). A survey on federated learning and its applications for

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 01
RECEIVED: JANUARY 07 REVISED: JANUARY 26 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 17

accelerating industrial Internet of Things. arXiv.

24. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Niyato, D., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(3), 1622–1658.
25. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Niyato, D., & Poor, H. V. (2021). Federated learning for industrial Internet of Things in future industries. IEEE Internet of Things Journal, 8(21), 16677–16693.
26. Nandan, B. P. (2021). Enhancing Chip Performance Through Predictive Analytics and Automated Design Verification. Journal of International Crisis and Risk Communication Research, 265-285.
27. Rahman, M. S., & Ghosh, T. (2023). Machine learning and Internet of Things in Industry 4.0: A review. Measurement: Sensors, 28, 100822.
28. Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., & Calinescu, A. (2022). Integration of artificial intelligence in the life cycle of industrial digital twins. IFAC-PapersOnLine, 55(10), 2545–2550.
29. The pipeline for the continuous development of artificial intelligence models—Current state of research and practice. (2023). Journal of Systems and Software, 199, 111615.
30. Deep reinforcement learning in smart manufacturing: A review and prospects. (2023). CIRP Journal of Manufacturing Science and Technology, 40, 75–101.