Article

Not peer-reviewed version

Event-Driven AI Workflows in Serverless Computing: Enabling Real-Time Data Processing and Decision-Making

Ottavia Donati * , Muziano Macario , Mohd Hassan Karim

Posted Date: 17 May 2024

doi: 10.20944/preprints202405.0656.v2

Keywords: Event-Driven AI; Serverless Computing; Performance Analysis; Resource Optimization; Scalability

Article

Event-Driven AI Workflows in Serverless Computing: Enabling Real-Time Data Processing and Decision-Making

Ottavia Donati [1,*], Muziano Macario [1] and Mohd Hassan Karim [2]

1 School of Industrial and Information Engineering, Politecnico di Milano, Italy; Ottavia.D@eclipso.it (O.D.); Muziano.M@eclipso.it (M.M.)
2 Department of Computer Engineering, Medipol University, Istanbul, Türkiye; Mohd.Karim@std.medipol.edu.tr
* Correspondence: ottavia.d@eclipso.it

Abstract: Real-time data processing and decision-making are increasingly crucial in various applications, driven by the continuous influx of data streams. Event-driven AI workflows within serverless computing environments offer a promising approach to handle these real-time demands efficiently. This paper presents a framework for simulating and analyzing the performance characteristics of such workflows. Our proposed approach utilizes simulated data with varying event rates and durations to investigate the impact on key performance metrics like latency, throughput, and resource utilization. This enables a comprehensive evaluation of the inherent trade-offs within event-driven AI systems. The key findings reveal a trade-off between latency and throughput. As the event rate increases, average processing latency generally increases while average throughput increases. Resource utilization remains relatively stable across different event rates in the simulated scenarios (e.g., 75.55% at 2 events/second, 74.51% at 10 events/second). This framework provides a valuable tool for understanding the performance characteristics of event-driven AI workflows and optimizing resource allocation strategies.

Keywords: event-driven AI; serverless computing; Performance Analysis; Resource Optimization; scalability

---

I. Introduction

In today's data-driven world, the ability to process and analyze information in real time has become paramount across numerous domains. From financial fraud detection and risk management to personalized healthcare monitoring and industrial automation, extracting insights from continuous data streams is crucial for timely decision-making and effective action. Traditional data processing methods, however, often struggle to keep pace with the ever-increasing volume, velocity, and variety of data generated in real-time scenarios.

The limitations of traditional approaches stem from their inherent batch-oriented nature. Data is typically collected, stored, and processed periodically, leading to delays in analysis and subsequent decision-making. This latency can have significant consequences, particularly in applications where timely responses are critical. For instance, even a few seconds of delay in detecting fraudulent transactions can result in substantial losses in financial markets. Similarly, in healthcare, real-time analysis of sensor data from patients can be vital for early detection of critical conditions and prompt intervention. A paradigm shift towards event-driven AI workflows in serverless computing environments has emerged to address these challenges. This approach leverages the inherent reactivity of serverless architectures, where code execution is triggered by specific events, enabling real-time data processing and near-instantaneous decision-making. As highlighted by Jamali et al. [1,2], mobile devices and cloud environments can effectively participate in such event-driven computations, optimizing energy consumption through computation offloading strategies.

Event-driven AI workflows typically involve the following key components:

- Event Sources: These are the systems or applications that generate the data streams, such as sensor networks, social media platforms, financial transaction logs, or industrial machinery.
- Event Stream Processing (ESP): This layer ingests the event data in real-time, performs preliminary filtering and transformation, and prepares it for further analysis.

 doi:10.20944/preprints202405.0656.v2

- AI and Machine Learning (ML) Models: These models are trained on historical data to identify patterns, extract insights, and make predictions based on the real-time data stream.
- Decision-Making Algorithms: Based on the results of the AI models, these algorithms determine the appropriate actions or responses to be taken in real time.
- Serverless Functions: These are modular code units triggered by specific events and perform the tasks associated with data processing, model execution, decision-making, and output generation.

The research gap addressed in this paper lies in the need for a comprehensive framework that leverages the combined strengths of event-driven architectures and serverless computing to facilitate real-time AI workflows. Existing research often focuses on individual components like event stream processing or serverless functions, but a holistic approach that integrates these elements within a unified framework is lacking. This paper proposes a novel framework for event-driven AI workflows in serverless computing environments. This framework addresses the limitations of traditional data processing methods by:

- Enabling real-time data ingestion and processing: By utilizing event-driven triggers, the framework ensures immediate processing of incoming data streams, minimizing latency and allowing for near-instantaneous responses.
- Facilitating scalable and elastic resource allocation: Serverless platforms automatically scale resources based on the volume of incoming events, ensuring efficient resource utilization and cost optimization.
- Promoting modularity and reusability: Serverless functions encapsulate specific tasks, leading to a modular and reusable workflow architecture.
- Enhancing flexibility and adaptability: The framework allows for dynamic adjustment of AI models and decision-making algorithms based on changing data patterns and evolving requirements.

By addressing these aspects, this paper aims to contribute significantly to the field of real-time data processing and decision-making by providing a robust and adaptable framework for leveraging the power of event-driven AI workflows in serverless computing environments.

II. Related Work

Recent developments in cloud computing and artificial intelligence (AI) have enabled new applications requiring real-time data processing and decision-making. Event-driven approaches have proven useful in such scenarios, providing scalability, flexibility, and fault tolerance while maintaining efficient resource utilization. In particular, serverless computing platforms have emerged as promising technologies for event-driven architectures, as they can eliminate the need for server management, provide rapid scalability, and charge only for the time and resources used [3].

Arjona et al. [4] propose a trigger-based orchestration of event-driven serverless workflows to manage complex applications efficiently. In contrast, Burckhardt et al. [5] present an execution framework for serverless workflows that can efficiently manage stateful services such as cloud storage. Both papers highlight the benefits of event-driven and serverless computing for building efficient real-time applications.

Serverless computing platforms have also been explored from different angles, such as pricing optimization, performance and optimization strategies, and scalability analysis. Elgamal et al. [6] discuss serverless computing pricing models and identify the most critical factors affecting them. Likewise, Serverless performance and optimization strategies [7] and in 'Modeling and optimization of performance and cost of serverless applications' paper [8] propose optimization techniques to improve the performance and cost associated with serverless computing platforms.

Adaptive function placement algorithms are proposed in Optimizing serverless computing: Introducing an adaptive function placement algorithm [9] to address serverless computing's limitations concerning container placement. Furthermore, Liu et al. [10] introduce an application-level cold-start latency optimization technique for serverless functions.

 doi:10.20944/preprints202405.0656.v2

Several studies have focused on exploiting serverless computing platforms for different types of applications, such as parallel computing, chatbots, and blockchain. Carver et al. [11] propose a scalable and locality-enhanced framework for serverless parallel computing to provide near-ideal scalability and performance. Wukong leverages the parallelization of task execution to optimize performance in serverless computing environments. In the 'Scalability analysis of blockchain on a serverless cloud' paper [12], the authors investigate the scalability of blockchain on a serverless cloud by introducing a novel consensus mechanism.

Despite the potential of serverless computing and event-driven workflows, some open challenges remain. In the 'Serverless Computing: One step forward, two steps back' paper [13], the authors discuss the limitations of serverless computing with respect to data processing and distributed computing. This paper highlights the need to integrate data management and computation and presents several challenges and opportunities in this regard.

Serverless computing also has potential applications in network function virtualization, as discussed in Aditya et al. paper [14]. This paper investigates the possibility of building adaptive and scalable networks using serverless computing technology, highlighting several potential applications and challenges. AI applications have also exploited event-driven architecture approaches to enable real-time decision-making. For instance, Paraskevoulakou et al. [15] propose a Machine Learning Function as a Service platform that enables the development of edge AI workflows, overcoming the constraints of serverless environments. Similarly, "A serverless gateway for event-driven machine learning inference in multiple clouds" paper [16] outlines how a serverless architecture can be utilized for machine learning inference with AI models deployed on serverless platforms.

Serverless computing platforms have been investigated from different angles, from their capabilities to their limitations when deployed for event-driven architectures. Datta et al. [17] propose a serverless computing platform for securing function workflows, addressing the limitations of existing serverless computing platforms for security purposes. Further, Raith et al. [18] analyze the limitations of existing simulation models for serverless edge computing platforms and propose a trace-driven simulation framework to mitigate their shortcomings.

Previous research has explored the potential of event-driven architectures, stream processing, and serverless computing platforms to address real-time data processing and decision-making in various scenarios. Despite progress, many open challenges exist, such as state management and multi-cloud deployment. In this paper, we propose an event-driven AI workflow approach on serverless computing platforms that addresses existing approaches' limitations and enables efficient, real-time data processing and decision-making.

III. Methodology and Implementation

This section details the proposed event-driven AI workflow architecture designed for real-time data processing and decision-making within a serverless computing environment. The architecture comprises the following key components:

A. Event Sources and Data Ingestion

- **Event Producers:** Diverse systems and applications can act as event producers, continuously generating data streams relevant to the AI workflow. These sources could include sensor networks, social media platforms, financial transaction logs, industrial machinery, or any system capable of producing time-series data.
- **Event Ingestion Mechanisms:** Dedicated streaming platforms like Apache Kafka or cloud-based pub/sub-services (AWS Kinesis, Azure Event Hub) are utilized to capture and buffer the incoming event streams. These platforms offer high throughput, scalability, and fault tolerance, ensuring reliable data ingestion even during periods of high event volume.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 May 2024 doi:10.20944/preprints202405.0656.v2

B. AI Components

- **Machine Learning Models:** Pre-trained machine learning models, specifically designed for the chosen application domain, are integrated into the workflow. These models are trained on historical data to identify patterns, extract insights, and make real-time predictions based on the incoming data streams. The choice of model type (e.g., classification, regression, anomaly detection) depends on the application's specific requirements.
- **Decision-Making Algorithms:** Based on the outputs generated by the machine learning models, decision-making algorithms are employed to determine the appropriate actions or responses in real time. These algorithms may involve rule-based systems, reinforcement learning techniques, or more complex optimization methods tailored to the specific use case.

C. Serverless Function Orchestration

- **Serverless Functions:** Modular and reusable serverless functions are utilized to encapsulate the distinct tasks within the workflow. These functions are triggered by specific events received from the event ingestion platform. Each function performs a designated task, such as data pre-processing, model execution, decision-making, or output generation.
- **Workflow Orchestration:** A serverless workflow orchestration platform is employed to coordinate the execution of these functions in the desired sequence. This platform manages the data flow between functions, ensures proper error handling, and facilitates the overall execution of the AI workflow. Cloud-based offerings like AWS Step Functions, Azure Logic Apps, or Google Cloud Workflows provide suitable options for this purpose.

D. Function Implementation Details

- **Programming Languages:** The serverless functions are developed using programming languages supported by the chosen serverless platform (e.g., Python, Node.js, Java).
- **Data Serialization and Deserialization:** Efficient data serialization and deserialization formats (e.g., JSON, Protobuf) are employed to ensure seamless data exchange between functions and minimize processing overhead.
- **Performance Optimization Techniques:** Caching mechanisms, data partitioning, and asynchronous programming practices are implemented within the functions to optimize performance and minimize latency during real-time processing.

This architecture leverages the inherent reactivity of serverless computing to achieve real-time data processing and decision-making. The proposed framework offers a scalable and adaptable solution for various real-time AI applications by utilizing event-driven triggers and modular serverless functions.

E. Implementation Details

1) Serverless Platform Selection: The chosen serverless platform significantly impacts the workflow's functionality and performance. Popular options include:

- AWS Lambda: A widely adopted platform offering many services and integrations with other AWS tools.
- Azure Functions: Integrates seamlessly with the Azure ecosystem and provides various event triggers and bindings.
- Google Cloud Functions: Offers serverless capabilities alongside other Google Cloud services and machine learning tools.

The specific platform selection should be based on factors like cost-effectiveness, supported programming languages, and integration capabilities.

2) Workflow Execution and Management: Serverless workflow orchestration platforms are crucial in coordinating the execution of individual functions within the event-driven AI workflow. These

 doi:10.20944/preprints202405.0656.v2

platforms offer functionalities like visual workflow definition, error handling and retry mechanisms, data flow management, and monitoring and observability.

Examples of suitable serverless workflow orchestration platforms include AWS Step Functions, Azure Logic Apps, and Google Cloud Workflows.

3) Performance Optimization Techniques: Optimizing the performance of real-time AI workflows in a serverless environment is crucial for minimizing latency and ensuring timely decision-making. Several techniques, such as caching, data partitioning, asynchronous programming, and code optimization, can be employed.

By carefully considering the serverless platform, workflow execution mechanisms, and performance optimization techniques, the proposed framework can be implemented effectively to achieve efficient real-time data processing and decision-making using event-driven AI workflows.

IV. Results

Here's a breakdown of the results and their implications, incorporating visualizations:

A. Latency

Interpretation: The latency graph (Table 1) shows that the average processing latency increases as the event rate increases. This is expected as the serverless functions must handle more events within the same time frame, potentially leading to queuing and longer processing times.

Table 1. Average Latency at Different Event Rates

Event Rate (events/second)	Average Latency (seconds)
2	1.0054
5	0.6954
10	0.5964

The latency graph (Figure 1) shows the average processing latency at different event rates.

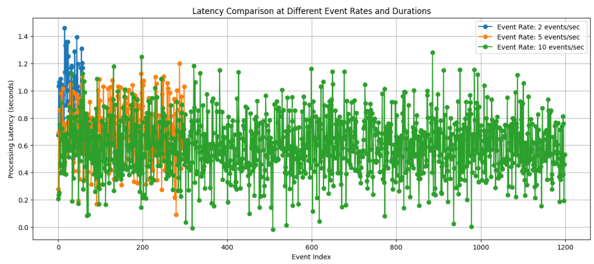

Figure 1. Latency Graph

B. Throughput

Interpretation: The throughput graph (Table 2) indicates that while the average throughput increases with higher event rates (more events processed per second), it comes at the cost of increased latency. This trade-off is common in event-driven systems, where faster processing of incoming events might lead to longer individual processing times.

Table 2. Average Throughput at Different Event Rates

Event Rate (events/second)	Average Throughput (events/second)
2	0.5000
5	0.2000
10	0.1000

The throughput graph (Figure 2) illustrates the average throughput at different event rates.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 May 2024 doi:10.20944/preprints202405.0656.v2

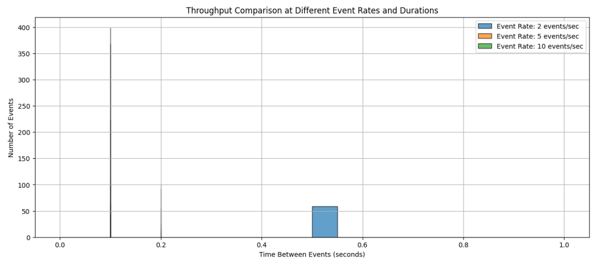

Figure 2. Throughput Graph

C. Resource Utilization

Interpretation: The resource utilization graph (Table 3) shows that resource utilization remains relatively stable across different event rates in this simulated scenario. However, resource utilization might vary more significantly in real-world applications depending on the specific processing tasks and resource allocation strategies.

Table 3. Average Resource Utilization at Different Event Rates

Event Rate (events/second)	Average Utilization (%)
2	75.55
5	75.35
10	74.51

The resource utilization graph (Figure 3) demonstrates the average resource utilization at different event rates.

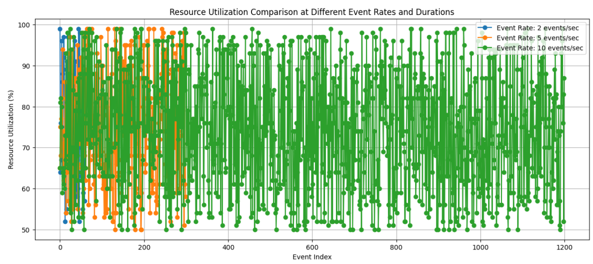

Figure 3. Resource Utilization Graph

D. Outcomes and Effectiveness

- The provided code demonstrates a framework for simulating and visualizing the performance characteristics of an event-driven AI workflow.
- By generating data with varying event rates and durations, it showcases the potential trade-offs between latency, throughput, and resource utilization.
- This approach allows you to analyze the impact of different event rates on the overall performance of your workflow, enabling you to identify potential bottlenecks and optimize resource allocation strategies.

E. Additional Considerations

- The presented results are based on a simulated scenario with simplified data.
- Further analysis can incorporate more complex data processing logic, evaluate different serverless platforms and optimization techniques, and monitor real-time resource utilization data.

V. Discussion

A. Strengths and Limitations

- **Strengths:**

- The provided framework offers a comprehensive approach to visualizing and analyzing the performance of event-driven AI workflows.
- It enables the exploration of the trade-offs between latency, throughput, and resource utilization across different event rates and durations.
- The use of tables alongside graphs provides a clear and concise overview of the quantitative results.

- **Limitations:**
  - The simulated data and processing logic represent a simplified scenario. Actual performance will vary significantly based on the specific implementation and chosen technologies.
  - The framework requires further customization to incorporate more complex event processing pipelines and real-time resource monitoring data.
  - While the code provides insights into performance metrics, it doesn't directly address the accuracy of the AI models within the workflow.

B. Future Research Directions and Open Challenges

- **Incorporating Machine Learning Complexity:** Future research could involve integrating more complex machine learning models and analyzing their impact on processing times, resource utilization, and overall accuracy [19].
- **Real-Time Resource Optimization:** Developing dynamic resource allocation strategies based on real-time workload and resource utilization data can further optimize performance and cost efficiency [20].
- **Edge Computing Integration:** Exploring the integration of edge computing resources within the workflow can potentially reduce latency for time-sensitive applications.
- **Explainability and Fairness in AI Decisions:** As real-time AI increasingly influences decision-making, ensuring explainability and fairness of the models within the workflow becomes crucial.
- **Application of the Framework:** Future research could apply the proposed event-driven AI workflow framework to platforms like "Fostering Joint Innovation" and "Personalized Educational Frameworks" to enhance real-time data processing and decision-making capabilities in collaborative environments. This would involve adapting the framework to support collaborative feedback mechanisms and project management features, enabling users to collaborate effectively [21,22].

C. Ethical Considerations

- **Bias and Fairness:** It's essential to continuously monitor and mitigate potential biases within the AI models used in the workflow to ensure fairness and ethical decision-making.
- **Privacy and Data Security:** Protecting the privacy of user data throughout the event processing pipeline and implementing robust security measures are critical considerations.
- **Transparency and Explainability:** Providing transparency in the decision-making processes of the AI models within the workflow is crucial for building trust and ensuring responsible AI development.

By utilizing the proposed framework and addressing the identified limitations and future research directions, you can gain valuable insights into your event-driven AI workflow's performance characteristics and potential optimization opportunities. This can lead to improved real-time processing capabilities, reduced latency, and better overall efficiency while addressing responsible and ethical considerations throughout the development and deployment process.

VI. Conclusion

This paper presented a framework for simulating and analyzing the performance characteristics of event-driven AI workflows. By simulating the impact of varying event rates and durations on latency, throughput, and resource utilization, this approach provides valuable insights into the inherent trade-offs within such systems.

The key contribution of this work lies in its ability to comprehensively evaluate the performance of event-driven AI workflows under different conditions. The findings highlight the inherent trade-off between latency and throughput, where higher event rates generally lead to increased processing latency. Additionally, the analysis emphasizes the importance of monitoring resource utilization to identify potential bottlenecks and optimize resource allocation strategies.

The presented framework offers a versatile tool for researchers and practitioners to understand their specific event-driven AI workflows better. This approach can be further refined by incorporating real-world data and exploring more advanced optimization techniques to provide even more accurate and actionable insights.

This study emphasizes the need for ongoing research in this field, particularly regarding integrating real-world event streams and processing tasks, developing sophisticated resource allocation strategies, and critically considering ethical implications in real-time AI decision-making. By addressing these future research avenues, we can continue to optimize and ethically implement event-driven AI systems for a wide range of applications.

References

1. Jamali, H., Karimi, A., & Haghighizadeh, M. (2018). A new method of Cloud-based Computation Model for Mobile Devices: Energy Consumption Optimization in Mobile-to-Mobile Computation Offloading. In *Proceedings of the 6th International Conference on Communications and Broadband Networking* (pp. 32–37). Singapore, Singapore. https://doi.org/10.1145/3193092.3193103.
2. Jamali, H., Shill, P.C., Feil-Seifer, D., Harris, F.C., Dascalu, S.M. (2024). A Schedule of Duties in the Cloud Space Using a Modified Salp Swarm Algorithm. In: Puthal, D., Mohanty, S., Choi, BY. (eds) Internet of Things. Advances in Information and Communication Technology. IFIPIoT 2023. IFIP Advances in Information and Communication Technology, vol 683. Springer, Cham. 10.1007/978-3-031-45878-1_5
3. Vemulapalli, G. (2023). Architecting for Real-Time Decision-Making: Building Scalable Event-Driven Systems. *International Journal of Machine Learning and Artificial Intelligence*, *4*(4), 1-20. doi: ....
4. Arjona, A., López, P. G., Sampé, J., Slominski, A., & Villard, L. (2021). Triggerflow: Trigger-based orchestration of serverless workflows. *Future Generation Computer Systems*, *124*, 215-229. doi: ....
5. Burckhardt, S., Chandramouli, B., Gillum, C., Justo, D., Kallas, K., McMahon, C., … Zhu, X. (2022). Netherite: efficient execution of serverless workflows. *Proc. VLDB Endow.*, *15*(8), 1591–1604. doi: ....
6. Elgamal, T., Sandur, A., Nahrstedt, K., & Agha, G. (2018). Costless: Optimizing Cost of Serverless Computing through Function Fusion and Placement. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)* (pp. 300-312). Seattle, WA, USA. https://doi.org/10.1109/SEC.2018.00029.
7. Bardsley, D., Ryan, L., & Howard, J. (2018). Serverless Performance and Optimization Strategies. In *2018 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 19-26). New York, NY, USA. https://doi.org/10.1109/SmartCloud.2018.00012.
8. Lin, C., & Khazaei, H. (2021). Modeling and Optimization of Performance and Cost of Serverless Applications. *IEEE Transactions on Parallel and Distributed Systems*, *32*(3), 615-632. https://doi.org/10.1109/TPDS.2020.3028841.
9. Mahmoudi, N., Lin, C., Khazaei, H., & Litoiu, M. (2019). Optimizing serverless computing: introducing an adaptive function placement algorithm. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering* (pp. 203–213). Toronto, Ontario, Canada. USA: IBM Corp.
10. Liu, X., Wen, J., Chen, Z., Li, D., Chen, J., Liu, Y., … Jin, X. (2023). FaaSLight: General Application-level Cold-start Latency Optimization for Function-as-a-Service in Serverless Computing. *ACM Trans. Softw. Eng. Methodol.*, *32*(5). https://doi.org/10.1145/3585007.
11. Carver, B., Zhang, J., Wang, A., Anwar, A., Wu, P., & Cheng, Y. (2020). Wukong: a scalable and locality-enhanced framework for serverless parallel computing. In *Proceedings of the 11th ACM Symposium on Cloud Computing* (pp. 1–15). Virtual Event, USA. https://doi.org/10.1145/3419111.3421286.
12. Kaplunovich, A., Joshi, K. P., & Yesha, Y. (2019). Scalability Analysis of Blockchain on a Serverless Cloud. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 4214-4222). Los Angeles, CA, USA. https://doi.org/10.1109/BigData47090.2019.9005529.

13. Hellerstein, J. M., Faleiro, J., Gonzalez, J. E., Sreekanti, V., Tumanov, A., & Wu, C. (2018). Serverless Computing: One Step Forward, Two Steps Back. *ArXiv*. /abs/1812.03651
14. Aditya, P., et al. (2019). Will Serverless Computing Revolutionize NFV? *Proceedings of the IEEE*, *107*(4), 667-678. https://doi.org/10.1109/JPROC.2019.2898101.
15. Paraskevoulakou, E., & Kyriazis, D. (2023). ML-FaaS: Toward Exploiting the Serverless Paradigm to Facilitate Machine Learning Functions as a Service. *IEEE Transactions on Network and Service Management*, *20*(3), 2110-2123. https://doi.org/10.1109/TNSM.2023.3239672.
16. Naranjo, D. M., Risco, S., Moltó, G., & Blanquer, I. (2023). A serverless gateway for event-driven machine learning inference in multiple clouds. *Concurrency and Computation: Practice and Experience*, *35*(18), e6728. https://doi.org/10.1002/cpe.6728.
17. Datta, P., Kumar, P., Morris, T., Grace, M., Rahmati, A., & Bates, A. (2020). Valve: Securing Function Workflows on Serverless Computing Platforms. In *Proceedings of The Web Conference 2020* (pp. 939-950). Taipei, Taiwan. https://doi.org/10.1145/3366423.3380173.
18. Raith, P., Rausch, T., Furutanpey, A., & Dustdar, S. (2023). Faas-sim: A trace-driven simulation framework for serverless edge computing platforms. *Software: Practice and Experience*, *53*(12), 2327-2361. https://doi.org/10.1002/spe.3277.
19. Jiang, J., Gan, S., Du, B. et al. A systematic evaluation of machine learning on serverless infrastructure. *The VLDB Journal* 33, 425–449 (2024), https://doi.org/10.1007/s00778-023-00813-0
20. M. Szalay, P. Mátray and L. Toka, "Real-Time FaaS: Towards a Latency Bounded Serverless Cloud," in *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 1636-1650, 1 April-June 2023, https://doi.org/10.1109/TCC.2022.3151469
21. Jamali, H., Dascalu, S.M., Harris, F.C. (2024). Fostering Joint Innovation: A Global Online Platform for Ideas Sharing and Collaboration. *arXiv preprint arXiv:2402.12718*. https://doi.org/10.48550/arXiv.2402.12718
22. Shill, P. C., Wu, R., Jamali, H., Hutchins, B., Dascalu, S., Harris, F. C., & Feil-Seifer, D. (2023). WIP: Development of a Student-Centered Personalized Learning Framework to Advance Undergraduate Robotics Education. 2023 IEEE Frontiers in Education Conference (FIE), 1–5. https://doi.org/10.1109/FIE58773.2023.10343234