APEX ACADEMIA PRESS
ELEGANCE ON EVERY RESEARCH

International Journal of Machine Learning, AI & Data Science Evolution
ISSN: 3068-5073/ Open Access / IJMLAIDSE_Y1F4A004
Apex Academia Press / https://doi.org/10.63665/ijmlaidse-y1f4a004

Original Article

Developing AI and ML-Based Real-Time Streaming Microservices for Distributed Systems

Faloye Oluwadamilohun Mary

Ladoke Akintola University of Technology, omfaloye@student.lautech.edu.ng

Abstract

Microservices architecture has made it easier for spread-out systems to change and grow. You can process and analyze even more data-even faster-by adding AI and ML to microservices streaming data in real time. This paper discusses the design and development of intelligent microservices, with the main focus being how to combine AI/ML with microservices architecture. We discuss various pros and cons of the use of AI and ML in microservices, especially when it relates to fetching data from different locations in real time. Such integration could be very useful in real-life applications, such as online shopping, doctor visits, or money-related dealings. The recommendations for further research and development in this domain conclude the paper.

Keywords

Microservices Architecture, Artificial Intelligence (AI), Machine Learning (ML), Real-Time Streaming, Distributed Systems, Intelligent Microservices, Data Processing, Scalability, Flexibility, Automation.

Article History

Received:
03.10.2025

Accepted:
12.10.2025

Published:
25.10.2025

1. Introduction

A. An Overview of the Architecture of Microservices

The entire microservices architecture consists of several services that need not tightly couple with each other; rather, each can run independently on their own. There are well-defined APIs which allow one service to communicate with another regarding the execution of a certain business functionality. This kind of architecture allows for creating, utilizing, and scaling services without depending on other services. In this way, it helps you become stronger, more flexible, and grow. Since all microservices are not located in a single place, it becomes easy to modify and deploy parts of the software quickly. This goes hand in hand with the concepts of continuous delivery and agile development.

B. Real-Time Streaming's Importance in Contemporary Applications

Applications that require observing and processing of data in real time have, more than ever, a growing need for real-time streaming. It's a fantastic capability to let your teams monitor systems, detect fraud, deliver personalized tips, and manage funds. With real-time streaming, businesses can become productive and keep users engaged by responding quickly to new trends, finding problems a lot earlier, and pushing the most current information to the user.

C. AI and ML's Contribution to Microservices Improvement

Adding ML and AI to microservices will enrich the system's capability to handle more complex data and make decisions. AI and ML models for decision-making can be used by microservices, including the prediction of what is going to happen, by monitoring data streams in real time. The integration is good for spread-out apps and systems since it will enable you to make use of predictive analytics, adjust how your app responds to new data patterns, and personalize each user's experience more effectively.

D. The Paper's Goals and Scope

This paper discusses changes in distributed systems over time and how to apply AI and machine learning in microservices running in real time. This will offer a comprehensive overview of the architectural concept, integration, and practical applications of smart microservices. Advantages and disadvantages in using AI and ML in microservices will be discussed; examples from several fields will be identified and ways found to conduct more research and development in this area.

2. Background and Related Work

A. Principles and Practices of Microservices Architecture

It makes designing easier because it breaks a system down into smaller, more manageable parts. Each service is operated by a different part of the company. These services teach people how to make, build, and grow things on their own, making them stronger and more adaptable. Some of the important concepts involve decentralized data management, continuous delivery, and isolating failures. The two common approaches toward building good microservices architectures are domain-driven design and lightweight communication protocols, such as HTTP/REST and messaging queues.

B. Integration of AI and ML in Software System

With the addition of AI and ML, software takes on algorithms that make computers learn from data, find patterns, and draw conclusions that can be proved. This link improves things like data analysis, predictive analytics, and natural language processing. You can also add models to microservices, get AI services with APIs, and add ML models in application codebases so as to work with the data and immediately make decisions based on it.

C. Current Approaches and Structures

AI and ML make it easier to build systems using microservices in many varied ways and frameworks. For example, the Lambda architecture leverages both batch and real-time processing to analyse a lot of data for thorough analysis with fast answers. Similarly, event-driven architectures help systems instantly act upon something that is happening in real time, crucial for applications that rely on processing and responding to data the moment it's captured. These frameworks let you build systems that can work with AI and ML and change and grow as needed.

D. Difficulties with Present Methods

Adding AI and ML to microservices can be difficult, but it can also be helpful. It is hard to keep data consistent, deal with the fact that ML models are stateful, and work with systems that are spread out. You may be concerned about the safety and privacy of your information if you are going to use it across several systems. A great amount of computing power is needed for AI and ML processes, an aspect that slows things down. To limit this, one has to know how to work with what exists. Planning of architecture, strong data governance procedures, followed by regular checking on the system and updating, are what need to be done to fix these problems.

This section describes, in general, what microservices architecture is, how AI and ML can be used in software systems, the frameworks that are already in place to help such integrations, and the problems that come up with current implementations. The rest of the sections discuss how to plan and stick to a plan, and this background makes that possible.

E. Developing Intelligent Microservices for Real-Time Streaming

Designing a system architecture that can handle continuous data streams with the latest features of ML and AI is what smart real-time streaming microservices require. It is this design style that makes real-time data much more responsive, scalable, and adaptable to help businesses derive useful information from it.

F. Aspects of Architecture

Real-time streaming microservices function best when their architecture is properly organized to meet requirements for high availability, low latency, and fault tolerance. An event-driven architecture is important because it will allow the system to respond quickly to events adding new data. Events in EDA tell the system to do things that let you work with data and make decisions right away. This way, each microservice can do its own job and talk to other microservices using simple protocols. This method makes it easier to grow and be flexible because services can be created, used, and added to without help.

G. Using ML and AI with Microservices

AI and ML turn microservices into smart parts that can handle and analyse incoming streams of data. Now, the link could be used by a microservice to make decisions, guesses, and suggestions. Microservices can use machine learning models for sending personalized content to people, determining when something has to be fixed, and finding fraud. This is an essential setup for low-latency inference because it facilitates easy processing of new data. Quantization and pruning are some means of enabling model performance in microservice architecture. These strategies facilitate real-time analytics through reductions in the amount of work to be done.

H. Mechanisms of Data Processing and Flow

In order to handle the enormous volume and velocity of real-time data streams, you need to be able to move and process data fast. Using a reliable data streaming platform makes it easier to send data to microservices for processing and share it across other microservices. These systems can handle events in a variety of ways such as event stream processing, simple event processing, and more complex types of event processing. They can handle simple events, as well as more complicated ones where they have to look for patterns across many different events. In stream processing frameworks, the system can utilize and learn from data immediately. This is what makes it possible to do analytics on data in real time. The system can meet the needs for real-time processing speedily and easily simply by reducing latency and increasing throughput through Data Flows. It can, therefore, react very fast against changes in data.

I. Providing Flexibility and Scalability

Smart real-time streaming microservices must be changeable and scalable because the system needs to adapt to changes in needs and loads. You can add more microservices since they don't remember what they did. To do more work without slowing down, you could add more instances. Containerization tools like Docker allow the microservices to uniformly have the same size for each situation, and they grow easily. Kubernetes and other orchestration tools will take care of placing apps into containers, making them bigger, and running them. They will automatically expand when a lot of people start using them or shrink when nobody uses them. Service meshes make interaction between services easier by adding features like load balancing, service discovery, and secure communication. They also make it easier for the system to grow and change.

3. Implementation Strategies

A. Choosing the Right AI and ML Models

Microservices work best once you choose the appropriate AI and ML models. Here, selection involves what an application does, what kind of data it would use, and what is to be achieved. Those that take a look at a chunk of data quickly and make a prediction in quick time are good for real-time streaming. Online learning models are good to keep whereby the model would continually get new data. One must find a balance in the model's complexity and its performance such that the data could be processed in real time without compromising accuracy. Testing and retraining of the models must be done quite frequently so that success may be maintained over time and cope with changing data patterns.

B. Integration with Streams of Real-Time Data

Data pipelines that can support a constant stream of data without bottlenecks have to be developed in order to connect the AI and ML models with real-time data streams. Such data streaming technologies are designed to transfer data in real time and process it straight away for fast actions and responses by humans. The microservices models will support incoming streams of data and can act based on the rules learned from experience or patterns. Most important of all, strong validation and preprocessing of data will be followed to ensure that the input ingested by AI and ML models is good and consistent to help avoid problems arising because of noisy or mismatched data.

C. Implementation in Dispersed Settings

In smart microservices, the challenges relate to service discovery, fault tolerance, and load balancing in a distributed system. Orchestration platforms help an individual monitor containerized services running on numerous nodes. It helps ensure the availability of services and resource utilization. Service meshes are instrumental in adding extra features such as load balancing, security, and observability to enable communication between services more conveniently. The service mesh also forms an integral part in ensuring the dependability and performance of a system. Deployment strategies for a system have to be planned in relation to handling faults and ensuring data consistency, considering network latency to enhance responsiveness and the reliability of the system under most conditions.

D. Practices for Monitoring and Maintenance

In the case of smart, real-time streaming microservices, there is a constant need to monitor and maintain them for proper running and reliability. Establish systems for comprehensive logging and monitoring that allow tracking system metrics and proactively find issues before things spiral out of control. Make sure warning systems are set up to ensure performance issues are resolved as quickly as possible. There will be minimal downtimes, and users' trust is secured. Testing and updating of AI and ML models from time to time is also an essential thing in confirming that their predictions remain accurate, for the simple reason that patterns change in data day in and day out. MLOps enables your teams to easily share and integrate ML models on a continuous basis. That can help these models stay in sync with business objectives and legislation, ensuring they keep delivering value over time.

In other words, with due considerations to design and implementation, scalable, adaptive, fast intelligent real-time streaming microservices can be built by organizations, providing insight into and action from the data in real time. It's not only a better way to make it all work; it's also a competitive differentiator with respect to decision-making on data.

4. Case Studies

Smart, real-time streaming microservices have redefined many industries by making the handling of streams of data and finding useful information from it much easier than before for an enterprise.

A. Sector of E-Commerce

E-commerce makes use of recommendation systems, which take into consideration people's actions and tastes in real time to suggest various products that will keep users engaged and drive more sales. Demand forecasting takes into consideration present information to make educated guesses as to how much of the product is needed. It helps you keep track of your inventory and minimizes the chances of running out of stock or having too much.

B. Finance Industry

Real-time transaction analysis can potentially allow banks and other financial institutions to monitor the instances of transactions, identify problems, and interdict fraud on the spot. Fraud detection systems monitor as transactions occur in search of anomalous behaviour- thus helping people keep their money.

C. The Healthcare Industry

Patient data monitoring systems capture and analyse data about a patient in real-time during patient care. This enables you to act fast and ensures that every patient receives optimal care. Predictive analytics of patient outcomes make predictions of what may happen in a person's health using current data. This will, therefore, finally enable you to plan your care effectively and optimize resources.

5. Problems and Fixes

Solutions are legion for different problems that arise during the creation and usage of smart real-time streaming microservices; each has its fix.

A. Data Security and Privacy Issues

A lot of effort is required to protect sensitive data against unauthorized persons and breaches: encryption, limitation of access, and observance of legislation on data protection are expected in order to maintain user confidence in your regard for privacy.

B. Handling the Complexity of the System

The bigger these systems are, the harder it is to track how many microservices have been implemented. Some of the tools that could make usage easier and reduce the complexity include centralized logging and monitoring, orchestration tools, and standardized communication protocols.

C. Providing Processing Capabilities in Real Time

Since we have to process everything in real time, we must implement architectures that will handle the data with low latency. We will be able to ensure high-speed processing of data and system response with event-driven architectures, an appropriate implementation of data streaming systems, and stream-processing frameworks.

D. Resolving Performance Obstacles

Profiling helps in the location and correction of problems that hamper the performance of a system, while runtime performance tuning involves periodic performance reviews, optimization of codes and data paths, and addition or removal of resources for optimum performance that will keep the system in its best condition and avoid bottlenecks.

However, if they knew how to resolve these issues, many businesses can employ smart real-time streaming microservices promoting innovation and operational excellence.

6. Prospects for the Future

Microservices architecture, coupled with machine learning and artificial intelligence, are running a new generation of distributed systems-smarter, more flexible, and more scaling than ever devised. Moreover, they constantly get better and open possibilities that were unimaginable so far. On the other hand, they make things harder and require lots of thought.

A. Developments in AI and ML Methods

Recent advancements in AI and ML have opened up new frontiers for microservices architecture. New ideas such as liquid neural networks, inspired by biological systems, perform well with temporal data, thus allowing better and more flexible models to be created. These networks will help you learn more and do better while things are changing. The application of AI and ML in network management truly creates smart and highly responsive distributed systems. It is also changing how the systems handle data traffic, utilize resources, and conduct fault detection.

B. Microservices Architecture Development

The architecture of microservices is constantly changing, with new application additions. SDA provides easier change and gives more options, since hardware and software are separated. This shift will make it much easier to build systems that can support more users and be more stable. The newest AI and ML features of these systems will also speed up decision-making and data processing.

C. Possible Fields of Study

Besides the advancement of microservices and the integration of AI/ML, several new areas become important. Standardization of the frameworks in MLOps will make it easy for microservices to deploy and manage ML models reliably. Building changeable systems that also cooperate requires learning how the various AI models can better interoperate across diverse microservices. Also, it would be of great help if the AI-driven decisions in a distributed system were made transparently to the people to gain their trust so that AI works correctly.

D. Effects on Distributed Systems Over Time

Time The use of AI and ML in microservices can bring a sea change to the working of distributed systems. This can result in smarter, more autonomous, self-optimizing systems which can perform predictive analytics and real-time decision-making. Such systems would manage resources better, be more failure-resistant, and adapt and learn from new situations with limited or no human intervention. At the same time, we will also have to consider moral implications of AI-driven behaviour in large groups, apart from ensuring data security and privacy.

7. Conclusion

Putting AI and ML together with microservices architecture is a big step forward in how people design, build, and use distributed systems. As this convergence goes on, it is bringing in systems which, besides being modular and scalable, would be smart, adaptive, and able to make decisions on their own. It changes how the technology works. Companies can exploit AI/ML combined with microservices in creating applications that will predict what users are going to need in the future, will make things work better, and change to meet the needs of users in real time without needing help from people all the time. These might revolutionize whole industries by offering predictive analytics, automated processes, and highly personalized user experience on an as-yet-unimaginable scale and efficiency. You are in charge of fixing any operational, ethical, or technical problems that come up during this integration. If you want these systems to be functional and fair, then you should be considering aspects like data privacy, model bias, security gaps, model drift, and system complexity. Further, we need to proceed with continuous research and development in this area to create monitorable and governable systems that can scale and robust DevOps and MLOps pipelines. AI and ML make microservices better, and more businesses use them. Software engineers, data scientists, ethicists, and system architects working across disciplines get the most from this architectural model. A combination of these technologies is likely to set up the next generation of smart applications and digital services able to run on their own. This will be a test for the distributed systems. In the future, microservices using AI and ML will be of great use. If they use new technology and have ethical design and running of their businesses, then they will be smart, efficient, strong, responsible, and in line with human values. What's happening is a revolutionary change, rather than one of degree, in how people think about software-its ability to learn, adapt, and change with the ever-changing and increasingly complex ecosystems it serves.

8. References

[1] Lucas Filho, E. R.; Savva, G.; Yang, L.; Fu, K.; Shen, J.; Herodotou, H. *Employing Streaming Machine Learning for Modeling Workload Patterns in Multi-Tiered Data Storage Systems.* Future Internet, 2025, 17(4):170. MDPI

[2] Dunning, T.; Friedman, E. *Streaming Microservices.* In: Sakr, S.; Zomaya, A. Y. (eds.) *Encyclopedia of Big Data Technologies.* Springer, Cham, 2019. SpringerLink

[3] Song, Chenghao; Xu, Minxian; Ye, Kejiang; Wu, Huaming; Gill, Sukhpal Singh; Buyya, Rajkumar; Xu, Chengzhong. *ChainsFormer: A Chain Latency-aware Resource Provisioning Approach for Microservices Cluster.* arXiv preprint, 2023. arXiv

[4] Fettes, Quintin; Karanth, Avinash; Bunescu, Razvan; Beckwith, Brandon; Subramoney, Sreenivas. *Reclaimer: A Reinforcement Learning Approach to Dynamic Resource Allocation for Cloud Microservices.* arXiv preprint, 2023. arXiv

[5] Mehran, Narges; Kimovski, Dragi; Prodan, Radu. *A Two-Sided Matching Model for Data Stream Processing in the Cloud-Fog Continuum.* arXiv preprint, 2021. arXiv

[6] Zhang, Hongyi; Bosch, Jan; Holmström Olsson, Helena. *Real-time End-to-End Federated Learning: An Automotive Case Study.* arXiv preprint, 2021. arXiv

[7] “Large-Scale Intelligent Microservices,” Mark Hamilton et al. *Deploying Machine Learning (ML) algorithms within databases…* Papers with Code, 2020. Papers with Code

[8] Sergio Moreschini; Shahrzad Pour; Ivan Lanese; Daniel Balouek-Thomert; Justus Bogner; Xiaozhou Li; Fabiano Pecorelli; Jacopo Soldani; Eddy Truyen; Davide Taibi. *AI Techniques in the Microservices Life-Cycle: A Systematic Mapping Study.* Computing, 2025. SpringerLink

[9] *Designing Microservices Using AI: A Systematic Literature Review.* MDPI, 2024. MDPI

[10] *Event-Driven Microservices: Building Responsive and Scalable Systems with Stream Processing*, Sanghamithra Duggirala & Ajay Shriram Kushwaha. Universal Research Reports, Vol. 12, No. 1, 2025.