International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.
IJSRA
International Journal of Science and Research Archive
Research Journal Archive, INDIA

(Review Article)

Serverless Data Pipelines for Scalable Financial ML Systems on AWS

Prasanth Sasidharan *

College of Engineering Trivandrum, Kerala.

International Journal of Science and Research Archive, 2025, 17(03), 387-394

Publication history: Received on 11 October 2025; revised on 18 November 2025; accepted on 20 November 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.17.3.3083

Abstract

Machine learning (ML) processes combined with serverless architecture, both on Amazon Web Services (AWS), have already proved to be a successful development of scalable, efficient, and cost-effective data pipelines, i.e., in the financial sector. The specified paper assumes the elaborate examination of the existing practices, frameworks, and approaches toward the implementation of serverless data pipelines to process financial ML systems. With the AWS services being used, i.e., Lambda, SageMaker, Glue, and Kinesis, financial institutions can now get real-time analytics, predictor models, and resource management on demand without incurring the operational overhead that they had to incur with traditional infrastructure. The architecture and the addition of ML, the characteristic of real-time analytics, the consideration of compliance, and the optimization of the cost of the serverless pipelines on the basis of AWS are critically checked. Moreover, it explores the idea of the resilience of the multi-cloud environment as well and points out the transformational aspect of AI in automated scaling and performance management. The article could be considered an illustration of the ways the implementation, functionality, and modifications of the financial ML applications in a cloud-native environment occur as a synthesis of existing literature and the market dynamics.

Keywords: Serverless Computing; Financial Machine Learning; AWS Data Pipelines; Cloud Scalability

1. Introduction

The greater combination of financial services with artificial intelligence (AI) has compelled the necessity to serve large quantities of data in an efficient, scalable, and cost-effective way. Banking institutions are creating a requirement to possess real-time forecasts to help in identifying frauds, credit rating, risk modelling, and algorithmic trading, which demands the speedy and unrestrained flow of information via processing systems. Serverless implementation to the services such as Amazon Web Services (AWS) has become a topical construct as well, to construct scalable, elastic, and efficient data pipes for machine learning (ML).

Like AWS Lambda, AWS Step Functions, AWS Glue, AWS S3, and AWS SageMaker, AWS is also a comprehensive platform enabling one to build and execute serverless ML pipelines without purchasing and operating servers. Serverless computing is cheaper to operate because it is on-demand and possesses the ability to scale overhead, with economic models of pay-per-use pricing, which particularly is convenient in highly loaded systems with sporadic demand, such as the financial sector. A growing number of financial streams of data flowing into the system are becoming more and more complex and time-sensitive, which means that serverless pipelines are a potential solution to offer real-time analytics and enable ML models to be continuously integrated to deliver actionable information. It is a report about the current state-of-the-art practices/techniques of implementing serverless data pipelines for scalable financial ML systems on AWS services.

* Corresponding author: Prasanth Sasidharan.

2. Architectural Design of Scalable Serverless Pipelines

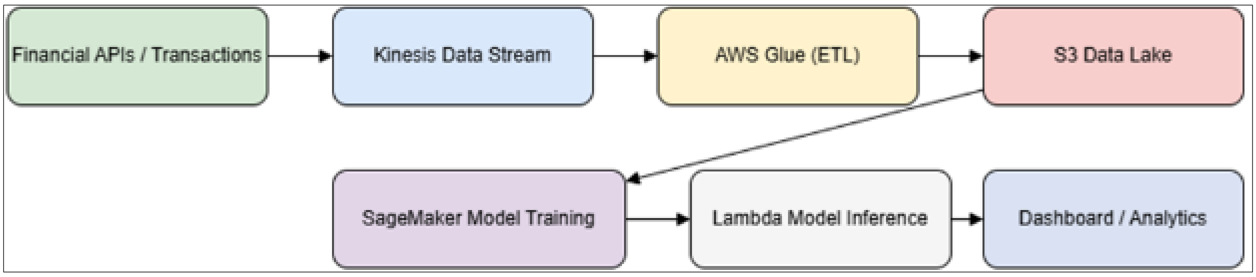
The four sequential phases the financial application scalable data pipelines comprise essentially entail ingestion of the data, transformation of the same, model training/inference, and saving/output of the data. AWS helps in these steps through the assistance of such services as Kinesis to support the stream of data ingest, Glue to offer ETL (Extract, Transform, Load) operations, SageMaker to control the model's lifecycle, and S3 to offer the scaled data storage.

The auto-process of the data after it is uploaded to S3 or the scheduled event by the conglomeration of the AWS Lambda is a critical response in architecture. This dynamic architecture allows financial institutions to build real-time pipelines that generally adjust to the quantity of incoming data in real-time without the involvement of persons and installation of servers. The temporary compute model being offered by Lambda is particularly applicable to micro-batch compute jobs (which micro-streaming analytics in industries tend to have in the financial sector) [1].

The Step Functions may be applied to organize workflows of Lambdas with only microservices performing all operations, which may include, e.g., data normalization, data cleansing, or feature engineering. As an illustration, an example of trading data can be accessed in a market by APIs and then uploaded by a Lambda function known as a data ingestion into an S3 bucket. The second Step Function will be in a position to invoke ETL work with the support of AWS Glue, which processes raw data into various formats that may be utilized in the analysis process. SageMaker transforms and retrieves data to be trained on predictive models or deployed endpoints to make an inference to either predict or deploy a product. The approach offers a continuous modularity of pipelines which is free of scales and may be scalable itself [1].

The selection of the most effective data flow strategies is one of the main points, which are to be raised within the context of the architectural design. AWS allows a serverless structure of stream processing and batch processing. Real-time processing and immediate feedback regarding the utilization of financial ML applications, in which latency is a significant factor, e.g., fraud detection or stock price prediction, can be obtained by stream processing with Amazon Kinesis Data Streams and Lambda and Firehose [1].

Figure 1 illustrates a high-level architecture of a serverless financial ML pipeline on AWS.



Source: [1]

Figure 1 Serverless Financial ML Data Pipeline on AWS

3. Machine Learning Integration in Serverless Pipelines

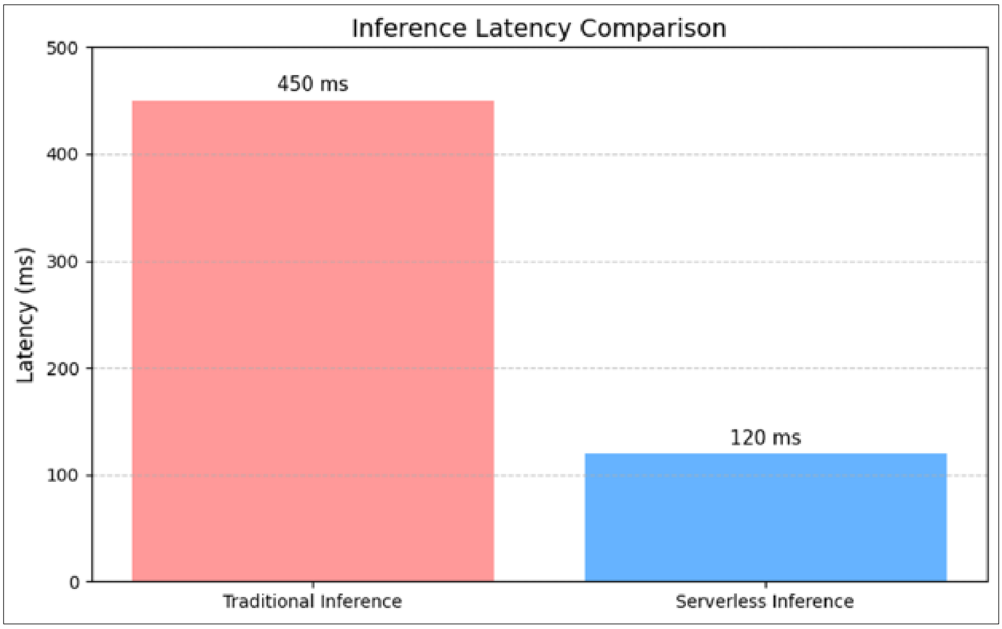
The automated decisions and predictions can be implemented in the financial systems by involving ML into the serverless data pipelines. AWS SageMaker will play a monumental role in the specified case as it will offer an entirely managed model training/tuning/deployment/monitoring environment. SageMaker will utilize multi-model endpoints, automatic model tuning, and A/B testing that will aid in the automation of the entire lifecycle of ML in a scalable manner.

An example of such patterns of integration can be to train a model using SageMaker offline and save it on S3, and predict using Lambda functions invoking SageMaker endpoints. This may be an architecture to support real-time scoring as well as offer low latency and high concurrency. It needs to be added that this way ensures that the training step and the inference one do not overlap in such a way that the loaded processes within the pipeline do not get hit by the resource-consuming training loads any longer [2].

The overall serverless ML pipeline will be structured as follows: pre-processing, through AWS Glue or Lambda, feature engineering, which is a sub-component of the ETL pipeline, and model inference, which is executed by lightweight models via SageMaker or Lambda. It implies that it can be programmed to experiment and retrain the model quickly in a continuous fashion without disrupting the accessibility of the production system due to this modularity [2].

Moreover, adaptive learning systems can also be obtained with the help of event-based Lambda functions, which can result in retraining of the models in case of data drift or a decrease in performance. It is particularly useful in instances when the financial market is a highly unstable phenomenon due to the swift rise or fall of the tendency of the regulatory processes, geopolitical processes, or consumer behavior [2].

Figure 2 shows a graph representing the latency comparison between traditional server-based inference and serverless inference pipelines.



Source: [2]

Figure 2 Inference Latency – Server-Based vs. Serverless ML Pipelines

4. Real-Time Analytics and Adaptive Pipelines

The necessity to investigate the volatility of the stock market, the work of credit, and real-time communication with their customers is causing additional pressure on the need to create real-time analytics of the financial system. When combined with Kinesis, Lambda, and DynamoDB, the applications with AWS serverless pipelines are the most applicable in real-time analytics applications. Through these services, it is possible to consume, transform, and query data in real-time many times within milliseconds, and financial ML systems can react to emerging data in real-time [3].

One of them is AWS Kinesis, which can access tick-by-tick stock exchange data and pass it onto Lambda functions to normalize and enhance the data. It is piped to an S3 bucket or Amazon Redshift or DynamoDB to work with a dashboard in real-time or score a model. The system allows financial analysts to make decisions within a few seconds based on data and is a source of competitive advantage and decreased vulnerability to risks [3].

The pipeline architecture of real-time ML workflows also requires feature stores. SageMaker Feature Store provides superior accessibility to both the training process and inference in pipelines. This will enable bringing the same data transformations as those in training to production, which will minimize the skew of the models and enhance the accuracy of the predictions [3].

The next factor is scalability. Scalability may be done automatically to manage the amounts of incoming financial data and provide constant performance without engaging additional resources. Stateless Lambda functions can be scaled to run in parallel and can be applied to serve events of millions per second [3], which is necessary in high-frequency trading algorithms or transaction fraud-detecting systems.

5. Efficiency and Elasticity in Cloud-Native Architectures

The data engineering frames are efficient and can be scaled to an enormous level that is also demanded to make the finance sensitive. Not only does stateless and event-based implementation of AWS Lambda eliminate idle-compute bills, but the pay-as-you-use billing also suggests that companies are only paying for what they are actually using. Besides that, AWS Glue and Athena offer on-demand data transformations and query services that reduce the number of resources utilized on the infrastructure and maximize the utilization of the resources to their full capacity [4].

The use of elasticity is also possible during high financial periods, like the opening of a trading session or even the presentation of economic data, in scenarios where the amount of data is large. Conventional server-based systems cannot support the unimaginable load changes and, as such, reduce performance. On the contrary, it is evident that the serverless services manage the scaling of resources depending on the demand, and it fails to cause broken performance even in times of the most substantial demand [4].

The Financial ML systems' batch inference job would be run with AWS Step Functions and AWS Lambda during non-peak time in order to maximize costs and achieve real-time predictions. In addition, serverless systems are easy to create and test and offer DevOps CI/CD, blue-green deployments, and canary testing. This allows the innovation process to be faster and enables the creation of new ML prototypes more quickly [4].

Table 1 summarizes the key differences between traditional and serverless ML pipelines in financial systems.

Table 1 Comparison between Traditional and Serverless ML Pipelines in Finance

Feature	Traditional ML Pipelines	Serverless ML Pipelines
Resource Provisioning	Manual	Automatic and event-driven
Scalability	Limited, requires planning	Automatic and elastic
Cost Model	Fixed, upfront	Pay-per-use
Latency	Higher, due to fixed resources	Low latency, fast spin-up
Maintenance Overhead	High	Low, managed services
Deployment Speed	Slower, manual interventions	Faster, integrated CI/CD

6. Secure and Compliant Data Handling in Financial Pipelines

Banking institutions are subjected to a thin layer of regulatory and compliance activities such as PCI-DSS, GDPR, and SOX. These require high data protection, data encryption, data auditing, and data access control throughout the path of the data. AWS is linked with many features in order to fulfill the requirements of a serverless architecture.

The AWS Lambda serverless application can be deployed into the Virtual Private Cloud (VPC), which will not be able to reach internal subnets and sensitive resources. This will guarantee that the services and information are separated, and this is very essential in dealing with financial information. The AWS Key Management Service (KMS) uses envelope encryption that can be applied as an indicator of encrypted sensitive data, which would be handled by a serverless element. The obtained information that has undergone transformation, logging processes, and inference procedures may be sent to Amazon CloudWatch and AWS CloudTrail to comply with audits and actions that have to follow the incidents [5].

Besides that, the security posture of serverless financial pipelines could also be improved with the help of the software of Amazon Macie (data classification) and GuardDuty (threat detection). Such technologies are based on the machine learning process to detect anomalies and unauthorized access to the system and are therefore used in the prevention of data breaches or misconfigurations in real time [5].

Most financial applications also demand the provision of fine-grained controlled access to servers with pipelines operated on a serverless basis. It is also possible through AWS IAM (Identity and Access Management) policies that role permissions could be allocated based on function, and all the segments of the pipeline could only access what they need for their operations. The principle of least privilege would reduce the size of the attack surface to the smallest extent and enhance the architecture to be as strong as possible [5].

Serverless designs also intensify the secure transmission of information between on-premise and cloud systems. This is quite factual, as the process of migrating old financial systems to cloud-native systems is still being undertaken. Introducing both AWS DataSync and Snowball allows for an easy encrypted transfer of information into the cloud store, i.e., S3, rendering it available to serverless ETL and ML services [5].

7. Model Performance and Predictive Accuracy in Cloud ML Pipelines

The performance and precision of models used in financial ML systems are highly sensitive, as a single deviation may have extremely massive economic consequences. Not only do AWS serverless pipelines streamline the procedure of data flow, but they also enable the monitoring of the performance of the high-level system to make predictions about the model. The automatic retraining features guarantee that the predictive force does not vanish in the long term.

The AWS SageMaker Model Monitor is one of the tools that might assist in the implementation of continuous monitoring of the deployed models to identify data distortion, feature bias, and degradation in the quality of predictions. With the assistance of Lambda and Step Functions, it is possible to initiate these monitoring alerts in the serverless pipeline architecture through retraining workflows. This will make sure the models applied in the market are able to adjust to new financial trends, policy changes, or any other locality in the market that is not expected [6].

Other than monitoring, there are distributed training and auto-hyperparameter optimization solutions by AWS, which are available under SageMaker. It can be optimized with the help of these capabilities and can be added to the pipeline even if it does not need to be done manually. An example of this is the automatic re-tuning of a financial credit scoring model on a monthly basis using up-to-date information on repayments made by customers. One of the scenarios in which a designed Lambda can be applied is when poor model behavior is detected, and training with new hyperparameters is initiated, followed by an evaluation and deployment of the model in case the new version fulfills the necessary accuracy requirements [6].

This type of automation can guarantee high-quality predictive behavior and financial ML system reliability. Besides that, S3 version control and SageMaker Pipelines datasets will also help to improve the reproducibility of training processes. Organizations can monitor the model development lifecycle using systems for maintaining training scripts, evaluation measures, and model binaries alongside the requirements for regulatory transparency [6].

It is possible to use such an automated retraining-versioning-performance-monitoring combination to make sure that serverless pipelines are able to provide steady model quality at any given time, even under the most erratic financial circumstances.

8. Resilience and Multi-Cloud Adaptability

Resilience is a critical requirement in financial implementation, as system downtime or outages may result in the violation of regulations or gigantic financial damages. Amazon serverless computing has an upper hand in ensuring that its services are resilient through fault tolerance, redundancy, and automatic retries. The AWS Step Functions service is a key example—one task failure will cause another restart, and Lambda functions can be programmed to use dead-letter queues to handle failures gracefully [7].

Moreover, the serverless pipelines may be set up to achieve high availability and disaster recovery due to the ability to use AWS-managed systems such as DynamoDB, which has backup, replication, and point-in-time recovery. Financial institutions are therefore able to ensure that their ML processes can continue even in the case of hardware, network, or local failures [7].

Financial institutions are migrating to multi-cloud architecture as well as hybrid cloud architecture as part of preventing vendor lock-in, improving resilience, and meeting data residency requirements in their operations. A multi-cloud environment that does not have serverless structures can also be serviced, and in this case, the event-based, loosely coupled design can be used. In one of them, AWS Lambda may be connected to other external APIs or systems within another cloud or on-premise, either by secure API Gateway connections or by AWS PrivateLink [7].

This is further improved by the fact that containerized serverless components can be installed in AWS Fargate or Lambda using the container image feature, wherein organizations can transfer their blocks of ML processing to new environments. Vendor-agnostic orchestrators, such as Terraform or orchestrators based on Kubernetes, can also coordinate serverless components [7].

Serverless architecture has been found to offer close to zero downtime, which is central to a financial ML system because of the problem of decoupling infrastructure, and it can be made available with high availability. Thus, it can be deployed to mission-critical systems.

9. Cost Optimization in Serverless Financial ML Pipelines

The banking companies are extremely price-sensitive, and they stand in need of infrastructural solutions that would allow them to gain cost visibility and cost control. At that, the AWS serverless systems can be described as good performers based on pay-as-you-go pricing schemes and automatic resource provisioning.

One such example is AWS Lambda, which will not incur idleness costs; instead, it charges based on time and milliseconds, which can be billed on the computation that is being executed. On the same note, other services are also available, such as AWS Glue, which can be billed on a per-job basis, and the cost also varies with the amount of data being processed; i.e., no permanent clusters or dedicated instances are necessary. These features allow a considerable amount of money to be saved compared to traditional VM-based infrastructures that are billed even in situations when the consumption is minimal [8].

Machine learning workloads — such as machine learning model training workloads and machine learning hyperparameter optimization workloads — are generally high-compute workloads. Spot Instances and Managed Savings Plans are available with AWS using SageMaker, which enables organizations to save up to 90 percent of training expenses without compromising performance. Moreover, it is possible to use SageMaker multi-model endpoints running on a single container to execute numerous models in order to achieve the lowest endpoint management cost and the highest utilization rates [8].

Cost optimization systems that are based on machine learning can also prove useful to enhance financial efficiency, such as AWS Compute Optimizer and AWS Cost Explorer. These tools can analyze usage patterns and suggest instance types, time schedules, or refactoring plans of the pipeline with the aim of reducing total costs. A typical report can propose the replacement of a Glue job with a more optimal Lambda-based conversion to reduce the overhead cost of the data pipeline, as an example of such an instance [8].

The use of performance and cost balancing can be applied in prediction modelling in financial applications. The selective allocation of high-performance resources in the serverless pipelines is only conducted where necessary. The fraud detection application has the capability to emphasize low-latency inference of a given SageMaker endpoint, and the portfolio optimization model has the capability of conducting training tasks in low-demand hours depending on the economic batch processing [8].

This has been done with regard to the flexibility that has enabled financial organizations to accomplish their analytical objectives without costly infrastructure.

10. Intelligent Scaling and AI-Driven Resource Management

This would require the scaling of resources in a dynamic, real-time financial environment, which has differing amounts of information. Auto-scaling in AWS can also be implemented without servers by default, and it can be augmented with artificial intelligence such that demand is predicted and pre-reserved.

One of them is the dynamic resource allocation of ETL pipelines, in which past trends can be employed by machine learning models to forecast workload properties. The models will be able to aid in increasing the scale of the AWS Glue job capacity or Lambda concurrency limit to absorb such patterns of predictable load. According to the case study, an artificial intelligence model could predict a continually expanding list of trading information at the time when a company is publishing quarterly earnings and allocate more resources so that the trading information would be available in sufficient time [9].

The objective of the serverless environment with AI-driven orchestration is to prevent over-provisioning and under-utilisation, with the purpose of identifying an improved cost-performance ratio. This is a very feasible and positive strategy in that any delays that could occur during the processing of the data can directly affect decisions made on investments, risk analysis, or compliance reporting [9].

Furthermore, the integrated predictive models that are combined and incorporated into CloudWatch and other variants of monitoring can assist organizations in producing self-healing pipelines. The precautions include preemption strategies such as concurrency management, re-execution of data flows, or execution of fallback models to detect anomalies in the execution time of functions, memory, or error rates — helping to restrain or trigger scaling [9].

Such kinds of operational intelligence are what will make serverless financial ML pipelines into reactive and proactive systems that can grow to become high-performing, reliable, and efficient without requiring the involvement of human beings.

11. Evolution of Serverless AI and Future Implications

Serverless AI is altering how development, deployment, and maintenance of ML applications ought to be done. The agility, speed, and creativity of this creation are massive in the financial sphere. Serverless platforms help data scientists to reason around the models and experimentation since they eliminate the infrastructure problem and do not require them to consider engineering.

Financial analytics expansion and its boundaries also lie in the advancement and improvement of features such as Serverless AutoML, federated learning, and edge-based inference. AWS SageMaker is able to collaborate in real time. It is possible to train multiple models and use distributed inference to enable teams to test the latest algorithms without necessarily deploying special teams in the ML process [10].

The broad-based applications of future serverless AI systems will hopefully be more business intelligence-enabled, providing real-time dashboards that directly receive the output of the models. This would imply that credit decisions, fraud warnings, and risk scores would be dynamically accessible to stakeholders within financial applications and be completely traceable and interpretable [10].

The serverless platform will also aid in further control of the models concerning the transparency, fairness, and accountability of AI. The default characteristics of serverless ML systems will include explainability, auditability, and versioning systems (such as SageMaker Clarify) that will manage operational workflows in compliance with regulations [10].

And finally, as serverless AI matures, it will provide a new paradigm of responsive, agile, and democratized ML systems, especially in some of the most tightly regulated areas of the financial sector.

12. Conclusion

The new AWS serverless paradigm has been created to offer financial machine learning systems efficient, scalable, and secure data pipelines. The services that allow financial institutions to train and deploy models without human intervention, maintain compliance, and reduce costs include AWS Lambda and SageMaker. Serverless architectures not only provide organizations with the strategic flexibility to react quickly to shifts in the market and regulatory environments, but they also provide the necessary technical capabilities. The concept of serverless AI is still in its infancy, yet the trend is emerging as a new standard for developing, evaluating, and deploying financial insights in the industry.

References

[1] Kukkamudi, S. (2025). Designing Scalable Data Pipelines with AWS: Best Practices and Architecture. Journal of Computer Science and Technology Studies, 7(9), 743-749.

[2] Thallam, N. S. T. Integrating Machine Learning into Big Data Pipelines: A Case Study with AWS SageMaker and EMR.

[3] Lawal, K. (2025). A Novel Framework for Next-Generation Data Pipelines in Real-Time Cloud Analytics.

[4] Singu, S. K. Serverless Data Engineering: Unlocking Efficiency and Scalability in Cloud-Native Architectures.

[5] Rohit, K. (2025). Agentic AI for Secure Financial Data Processing: Real-Time Analytics, Cloud Migration, and Risk Mitigation in AWS-Based Architectures.

[6] Chathurpally, A. G. (2025). Enhancing Predictive Analytics through Machine Learning Models in Cloud Computing Environments (Doctoral dissertation, Dublin, National College of Ireland).

[7] Manukonda, A. K. (2025). Multi-Cloud Serverless Computing & FaaS Architectures for Resilient and Cost-Efficient Systems. International Journal of Emerging Trends in Computer Science and Information Technology, 107-125.

[8] Mahimalur, R. K. (2025). Optimizing AWS Costs With Machine Learning-Driven Recommendations. Available at SSRN 5282647.

[9] Rodrigues, D. N., Rosas, F. S., & Grácio, M. C. C. (2025). Dynamic Resource Allocation in Serverless ETL: AI-Driven Scaling and Cost Optimization Models.

[10] Gupta, S. (2025). The Rise of Serverless AI: Transforming Machine Learning Deployment. European Journal of Computer Science and Information Technology, 13(5), 45-67.