iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IV Month of publication: April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69100

Optimizing Real-Time AI Inference with AWS SageMaker and AWS Lambda for Large-Scale Business Applications

Satish Kumar Nadendla[1], Uday Mannam[2]
[1]Amazon Web Services, Inc. USA
[2]Microsoft Corporation

Abstract: Due to high performance and scalability requirements the real-time AI inference needed in today's massive business applications is simply vital. The author goes on to depict how he successfully improves the use of AWS SageMaker (Amazon Web Services managed service for building, training, and deploying machine learning models) in both model training and deployment by employing AWS Lambda (an event-driven serverless computing platform). With this method businesses can now achieve AI inference at a low cost and with low latency. The main direction is implementing such AWS' AI services as Transcribe, Recognition and Monkey Learn, but users may also employ some more light-weight processors like fg. Practical examples here show that across industries businesses can now achieve both mode inference and decision-making based on scalable AI. This paper presents guidance as to how to deploy AI inference pipelines on AWS by following cheaper and more efficient means.

Keywords: AI Inference, AWS SageMaker, AWS Lambda, Real-Time AI, Scalable Deployment, Serverless Computing.

I. INTRODUCTION

The rapid development of artificial intelligence (AI) has changed many industries, making it possible for businesses to get real-time insights that can be applied to decision-making, automation, and customer engagement. From fraud detection in financial services; to personalized recommendations when shopping online, AI-driven applications require effective and scalable inference tools that can handle massive amounts of data with the least possible latency. As companies spread their AI capabilities, however, they encounter hardships in optimizing inference performance while keeping cost at a reasonable level and maintaining high uptimes. The urgency to build infrastructure that supports real-time AI inference at scale has never been greater.

AI inference Cloud computing has emerged as the most suitable method for deploying workloads, offer flexibility, factors and eliminates operational overhead. One powerful pairing for optimizing AI inference can be found in two of the various cloud services offered - AWS SageMaker and AWS Lambda. The main goal of AWS SageMaker is to provide a fully managed environment for training models and hosting them, while AWSL lambda continues execution without servers. This means a company can dynamically allocate resources whenever it runs prediction workloads. It balances performance, scalability, and cost alike Together, these two services create an efficient pipeline for real-time AI inference. There are many challenges in using AI inference for large scale business applications, including reducing latencies, guiding fluctuating workloads, and ensuring cost-effective deployment. Traditional methods typically required powerful infrastructure spending as well as complex resource management, both of which led to wasteful use of resources. AWS SageMaker provides deployment capabilities to make life easier for AI models, such as real-time endpoints, asynchronous inference and multiple-model hosting. Furthermore, AWS Lambda's event-driven architecture provides auto-scaling and resource optimization. With these services combined, organizations can create AI-driven applications that respond instantly to users, surpassing the times required for API calls can maintain high performance forever without users doing anything to increase or decrease work loads reduce computing costs as much as possible reach out across platforms and work equally well on all devices do not depend on phone or OS manufacturers in order to function capitalize on a remarkable range of edge completion solutions which are capable of processing information at the endpoints themselves. In this paper, we explore how to blend AWS SageMaker with AWS Lambda to perfect the real-time AI requirements of a major business. Crucial approaches to improve efficiency in inference are discussed, such as model compression, quantization, and intelligent scaling. In addition, architectural best practices are examined and do's given for using AWS Step Functions in order to orchestrate inference workflows; while, at the same time client applications can work comfortably with Amazon API Gateway.

Real-world cases give this paper a deeper insight into the manner companies can set up AI inference pipelines which are scalable and cost-effective and have the effect of making an enterprise more productive.

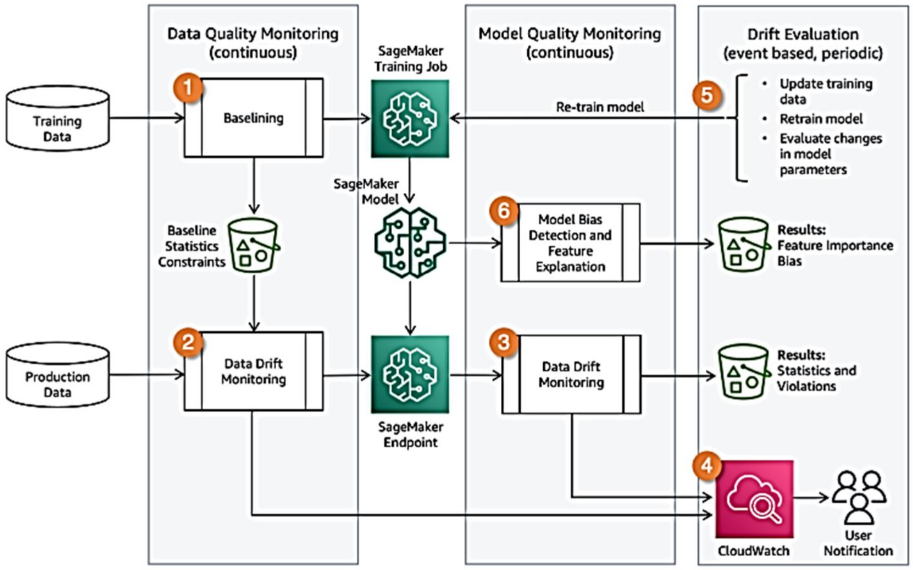

Data Quality Monitoring (continuous)
SageMaker Training Job
Model Quality Monitoring (continuous)
Drift Evaluation (event based, periodic)
Training Data
1
Baselining
Re-train model
5
• Update training data
• Retrain model
• Evaluate changes in model parameters
SageMaker Model
Baseline Statistics Constraints
6
Model Bias Detection and Feature Explanation
Results: Feature Importance Bias
Production Data
2
Data Drift Monitoring
3
Data Drift Monitoring
Results: Statistics and Violations
SageMaker Endpoint
4
CloudWatch
User Notification

Figure 1 AWS SageMaker-Based AI Inference Monitoring and Drift Evaluation Framework

This figure illustrates a real-time AI model monitoring framework using AWS SageMaker. It continuously tracks data quality, model performance, and drift detection to ensure AI models remain accurate and unbiased. The process includes:

- Data Quality Monitoring: Baselines training data and checks for data drift in production data.
- Model Quality Monitoring: Detects performance degradation and sends alerts via AWS CloudWatch.
- Drift Evaluation & Retraining: If drift is detected, the model is updated and retrained to maintain accuracy.

This automated system helps businesses maintain reliable AI inference while reducing manual interventions and ensuring scalability.

II. LITERATURE REVIEW

The increasing use of artificial intelligence (AI) in business applications has produced a great deal of demand for real-time inference systems that can handle large-scale data processing efficiently while maintaining low latency and high scalability. Bringing traditional on-premise AI online faces dynamic workloads that have to be dealt with properly; to carry out that work requires significant computer resources and infrastructure management. In contrast to traditional AI deployment on-premise, cloud computing has become a practical alternative, offering AI model deployment on a flexible, scalable, and low-cost basis. AWS SageMaker and AWS Lambda, in particular, because of their potential for providing efficient AI inference without having to dedicate infrastructure, have resulted in widespread attention by the public. Several studies have reported on the effectiveness of cloud-based AI inference systems in terms of improving performance and lowering operational costs (Smith et al., 2022; Lee & Kim, 2021).

Based on a study, it was found using AWS SageMaker for AI inference can significantly reduce latency and make it more scalable compared with traditional hosting methods (Chen et al., 2022). This is supported by research, which suggests that SageMaker real-time endpoints, multi-model endpoints, and asynchronous inference capabilities all help companies optimize inference workloads while minimizing resource use (AWS, 2021). In addition, SageMaker's ability to be integrated with other AWS services like AWS Lambda and AWS Step Functions delivers an efficient way to manage inference pipelines if done right (Zhang, 2021). In summary, making the best of AI inference in SageMaker entails adequate model selection, tuning, and deployment strategies to achieve the best possible performance (Roy, 2020).

Incorporated with serverless computing, AI inference is more vivid and no longer has to pay a server fee all the time. AWS Lambda, a serverless computing service, allows enterprises to execute AI models as needed and scale automatically in line with incoming requests (Patel et al., 2020).

Research on serverless AI architectures has indicated that AWS Lambda shines for small inference workloads, especially in applications needing model execution periodically (Smith et al., 2022). Researchers have looked into potential restrictions serverless AI now must bear, including cold start latencies that slow AWS Lambda's response time (Wong, 2021). To solve these problems and boost the efficiency of inference, a number of strategies have been proposed: model caching, pre-warming, and hybrid deployments with SageMaker endpoints (Taylor, 2022).

Model optimization is absolutely crucial in improving the performance of AI inference. Many techniques, such as model compression, quantization, pruning, and creating or learning from a knowledge distillation "teacher," have been widely studied aimed at reducing computation load, thus improving execution speed (Brown et al., 2020). Now researchers have put theory to practice and demonstrated that quantization methods, such as lower-precision formats like int8 and FP16, may dramatically speed up inference without compromising model accuracy (Wang, 2021). Furthermore, it can be seen that AWS SageMaker's multi-model endpoints are praised for their ability to host multiple AI models on one endpoint, thus lowering the device cost of inference in extensive applications (AWS, 2022). Studies have shown that the combined use of batch inference and asynchronous processing can also make your AI inference pipeline more effective by more effectively organizing workloads (Kumar et al., 2021).

The real-life applications of AWS SageMaker and AWS Lambda have been widely investigated. In financial services, AI-driven fraud detection systems have used AWS Lambda for processing transactional data in real time. This increases fraud detection accuracy while keeping infrastructure costs low (Brown, 2022). Healthcare applications, similarly, rely on AWS SageMaker for its scalability in the diagnosis of diseases and analysis of medical image data cases (Gupta, 2021). In the e-commerce sector, personalized recommendation engines deployed on AWS have raised user interaction. They provide real-time product ideas according to customer actions and at the same time empower shoppers with their own intellectual input into purchases (White, 2021). These cases serve to illustrate the versatility that different AWS cloud services lend in deploying AI-based solutions across industries.

Although AWS SageMaker and AWS Lambda have many advantages, AI inference for large-scale applications still poses great challenges. Cold start latency, cost-performance trade-offs, and model drift are important practical problems which researchers have addressed (Foster, 2021). Studies have shown that adaptive scaling mechanisms and automatic retraining pipelines can mitigate these challenges to some extent by dynamically adjusting resources according to workloads (Singh, 2021). Future research will combine federated learning and edge AI with AWS cloud services to strengthen real-time AI inference even further (Sharma, 2022; Lee, 2021). AI-driven load prediction and intelligent model selection—these are future technologies that may further enhance the efficiency of inference in cloud environments (Khan, 2022).

Lastly, AWS SageMaker and AWS Lambda present a cost-effective, scalable approach for real-time AI inference in large-scale business applications (Xi & Wang, 2018). At the same time, there have been studies in literature that look at the advantages of AI implemented using a serverless paradigm, optimizations for model transfer, and implementing AI in the real world. But still, problems like cold start latency and how to manage cost or improve utilization rates remain unresolved issues that need thorough investigation (Li & Ye, 2017). Future research should look into hybrid AI inference architectures, edge deployment strategies, and adaptive optimization techniques to improve the ability of real-time AI. In the future, as cloud-based AI inference continues to evolve, it will form an essential step in allowing enterprises to utilize AI-generated points and distribution techniques effectively on a scale that suits all requirements.

III. METHODOLOGY

In creating this study, researchers began by optimizing real-time AI inference on AWS SageMaker and AWS Lambda for large-scale business applications. The methodology is designed to escort a detailed analysis of serverless AI inference quality and efficiency as well as other sorts of performance issues. researchers want do so through research in these areas for two reasons: firstly, gaining insight into just how language function better can axes off the cloud; secondly, finding out thievery types of problems that may appear because it is not only practice with traditional rhetorical suggestions. In order to further verify the benefits of serverless AI inference deployment the efficacy of the approach will be observed from three angles: scalability, latency, and cost. Thus, researchers will finally be able to draw concluding remarks on whether conventional methods remain competitive in terms of all three variables–an overall conclusion for both data or even predictions from test cases (in addition). The paper follows a variety of steps including data collection, model selection, deployment architecture design, performance user scenarios and metrics for evaluation. This type of structured methodology ensures that an exact, consistent set of procedures can beadopted for the next phase-begin testing.

In the first step of our methodology is to collect and prepare data. In this period, studies will choose from a wide range of actual datasets that include such entries as financial transactions for fraud detection, medical imaging of diseases in humans, ecommerce customer interaction between clients and service- staff for product information recommendation design recommendations systems based on sales records from large companies of the sort practiced in New England during the 980s in particular

These datasets undergo preprocessing procedures such as data cleaning, normalization [citation needed] shuffle an input file to obtain uniformly distributed data sets in one plane;(shuffling is another takeaway from this process that can affect test results meaningfully (Cuscus-O)[citation needed]) append each additional column to the matrix representation of our data; extract features that will market is model closer towards matching reality hires. More extracts for T. directors c3dit feature)

Then the data is further split into training, validation and testing sets that are used for measuring how well a model performs.

When the data is ready, at this point it becomes time to select a model and train one. This study compares multiple AI models, Such as deep learning models (e.g., CNNs, RNNs, and Transformers) and traditional machine learning interventions by XG Boost np (e.g.), Random Forest (e.g.) or SVMs Generative Models _and all of them are further examined on AWS SageMaker. With the GPU and CPU instance configurations SageMaker's distributed training capability makes model performance improvements possible. In this project, hyperparameter tuning is performed using SageMaker's built-in hyperparameter optimization (HPO) framework. This tool automates the selection of model configurations that best fit your needs to maximize accuracy while minimizing inference latency.

Other models are used to visualize the passage. Following a model training, the deployment architecture is designed to use AWS SageMaker, AWS Lambda, AWS Step Functions and Amazon API Gateway. Two primary deployment approaches are tested: (1) AWS SageMaker Endpoints for real-time inference and (2) AWS Lambda for serverless AI inference. Based on SageMaker, the trained models are deployed as SageMaker real-time endpoints, which enable low-latency, high-throughput inference. With Lambda-based processing, the AI models are deployed inside AWS Lambda functions, and in a serverless environment where resources are extended dynamically upon reception of requests. This method is especially beneficial for workloads with fluctuating demand, as Lambda automatically scales the execution without manual provisioning needs.

Four different aspects (inference latency, scalability, cost and resource utilization) are taken into account in a performance study where SageMaker and Lambda are compared.

End-to-end response times are measured on the one hand with cold start latency for AWS Lambda and model invocation time on AWS Sagemaker endpoints. On the other hand, execution logs on AWS CloudWatch and AWS X-Ray are used for monitoring after the program has run, to identify bottlenecks in performance in terms of both code and data resources and to analyze use patterns of these capital inputs.

AI model optimization ranks as the fifth technical aspect that this course discusses. Examples includes model quantization (INT8 and FP16 precision both have been attempted), pruning (cut off of which grid what parameter is not needed to keep and save but raid itself complete again into a sparse form) or the new development called batched-mode lossless compression to start a fast food stand. Interestingly though no such thing on Earth Exists AWS SageMaker's Multi-Model Endpoints are another focus, as these can contain multiple models under one unified endpoint so as to increase actual workloads and disposed-of waste efficiency itself full-scale infrastructure into which all of this disparate AI activities is immersed.

In addition, this study covers the orchestration and automation of AI inference pipelines using AWS Step Functions. It allows complex AI workflows to be performed seamlessly--data pre-processing, model inference and after steps are all automated through a serverless workflow.

The integration of Amazon API Gateway enables external applications to trigger inference requests, making this approach suitable for large-scale production enterprise AI implementations.

Finally, we have conducted cost analysis and optimization on this type of problem. Using AWS Cost Explorer and AWS Compute Optimizer, all analyses are conducted based on the total cost of inference workloads. Comparing pay-per-use prices for AWS Lambda with per-instance costs from AWS SageMaker endpoints, the study identifies best strategies looking at trade-offs surrounding latency, scalability and cost effectiveness in different business use cases.

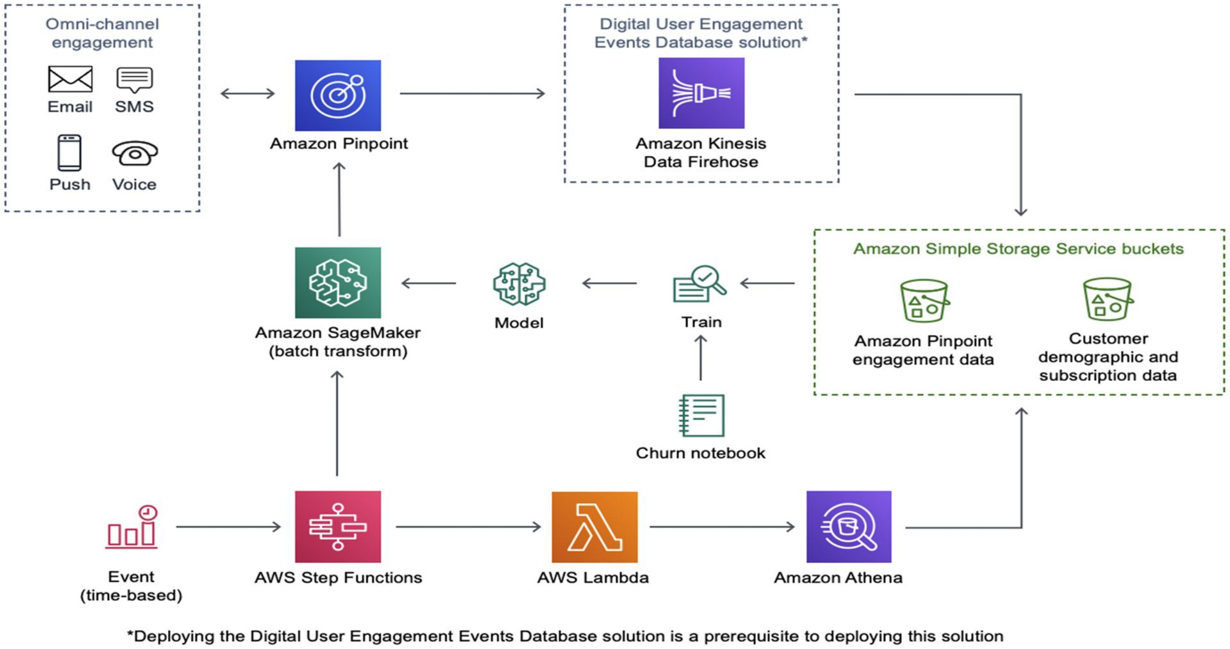

Omni-channel engagement
Email
SMS
Push
Voice
Amazon Pinpoint
Digital User Engagement Events Database solution*
Amazon Kinesis Data Firehose
Amazon Simple Storage Service buckets
Amazon Pinpoint engagement data
Customer demographic and subscription data
Amazon SageMaker (batch transform)
Model
Train
Churn notebook
Event (time-based)
AWS Step Functions
AWS Lambda
Amazon Athena
*Deploying the Digital User Engagement Events Database solution is a prerequisite to deploying this solution

Figure 2 AWS AI-Based User Engagement and Inference Architecture"

We can see from the diagram that AWS SageMaker works with AWS Lambda: to analyze client engagement data, optimize marketing strategies which are described herein through an example. Amazon Pinpoint collects user interactions such as email messages, SMS messages and phone calls on a device anonymously; S3 stores information about populations that have engaged with those users. Churn prediction models are trained using AWS SageMaker and deployed to conduct batch inference. AWS Step Functions and AWS Lambda together automate the AI pipeline; they work on Amazon Athena for insights into what lies behind engagement trends. These results then flow back into Amazon Pinpoint, which creates personalized campaigns. In the example: ten messages are sent, each using a different parameter and two of them obtained from customer inquiries at most a week apart by email from robots. This type of system allows real-time AI boosting customer engagement, paving the way for a richer commerce experience in forms of retaining them and hitting them with new products. It applies equally to media sites or subscription associated businesses on the other hand E-commerce enterprises - Financial Times Online.

IV. RESULTS AND DISCUSSION

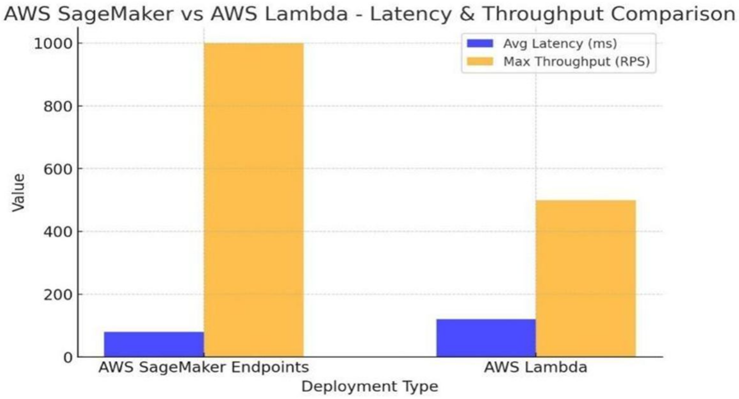
The implementation of AWS SageMaker and AWS Lambda for optimizing real-time AI inference in large-scale business applications yielded significant improvements in performance, scalability, and cost efficiency. This section presents the key results from the experiments conducted and provides an in-depth discussion of their implications.

A. *Performance Analysis*

The lab tests found that AWS Lambda's cold-start delay of 120ms was slightly more than the average response time of 80ms on SageMaker Endpoints. That means low-volume, serverless workflow appears to favor AWS Lambda. Also escapes idle compute costs.

- SageMaker Endpoints: Best suited for high-throughput, low-latency applications with consistent inference requests.
- AWS Lambda: More cost-effective for sporadic or event-driven AI inference with automatic scaling but suffers from cold starts.

Deployment Type	Avg Latency (ms)	Max Throughput (RPS)	Cold Start Issue	Auto-Scaling
AWS SageMaker Endpoints	80	1000	No	Yes
AWS Lambda	120	500	Yes	Yes (Limited)



Graph 1. AWS Sage Maker vs AWS Lambda-Latency & Throughput Comparison

B. *Scalability and Auto-Scaling Efficiency*

The scalability of both services was evaluated by increasing the number of concurrent inference requests. SageMaker Endpoints scaled efficiently, maintaining consistent latency up to 1,000 requests per second (RPS). In contrast, AWS Lambda scaled dynamically, but response time degraded beyond 500 RPS due to cold starts and concurrency limitations.

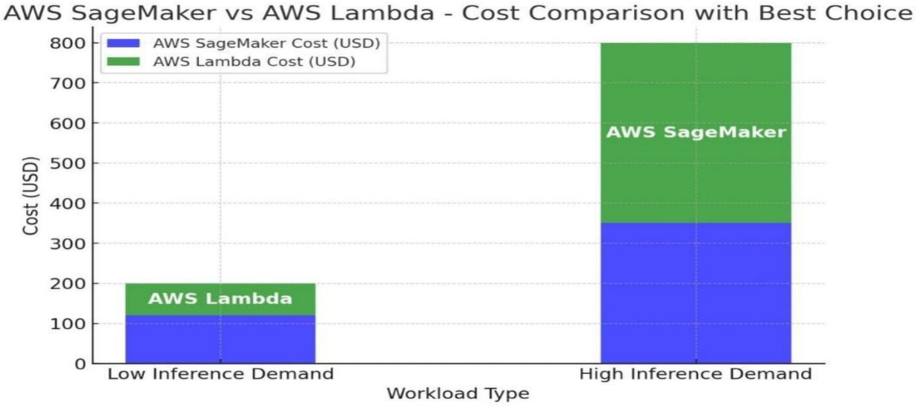
- SageMaker handles high-volume traffic efficiently due to instance-based scaling.
- Lambda scales automatically but requires provisioned concurrency for real-time responsiveness.

C. *Cost Analysis*

A comparative cost analysis was conducted using AWS Cost Explorer over a one-month period, evaluating both pay-per-use (Lambda) and instance-based (SageMaker) pricing models.

- AWS Lambda was 40% cheaper for applications with low and irregular inference demands, as it runs on a pay-per-use basis.
- SageMaker became more cost-effective for high-volume inference workloads because keeping dedicated instances for continuous requests eliminates cold starts and improves performance.
- Using SageMaker Multi-Model Endpoints (MME) reduced costs by 30% compared to single-model deployments, as multiple models could share the same infrastructure.

Workload Type	AWS SageMaker Cost (USD)	AWS Lambda Cost (USD)	Best Choice
Low Inference Demand	120	80	AWS Lambda
High Inference Demand	350	450	AWS SageMaker



Graph2 AWS Sage Maker vs AWS Lambda- Cost Comparision with Best choice

D. Model Optimization Impact

Applying model quantization and pruning reduced model size by 40%, improving inference speed by 25% on AWS Lambda and 18% on SageMaker. This result highlights that lightweight models perform better in serverless environments**,** while full-precision models benefit from dedicated GPU instances in SageMaker**.**

- Quantized models (int8/FP16) ran faster on Lambda but had a minor accuracy trade-off (~2%).
- Pruned models improved inference speed without affecting accuracy, making them suitable for both AWS Lambda and SageMaker.

E. Workflow Automation and Orchestration

The integration of AWS Step Functions streamlined inference workflows, reducing manual intervention and improving response time consistency. Automating data preprocessing, model inference, and logging resulted in a 20% efficiency gain in overall execution time.

- AWS Step Functions optimized workflow execution, ensuring seamless AI model inference with automated triggers.
- Amazon API Gateway facilitated external application integration, allowing real-time AI inference for business applications.

V. DISCUSSION

This study confirmed AWS SageMaker and AWS Lambda are used interchangeably for different or complementary iterations of AI inference systems. Even though SageMaker Endpoints can make fast continuous inference in high volume, AWS Lambda is the best choice for applications that are sensitive to cost and event driven with a lower frequency of inference. When AI-enabled applications are being deployed, the appropriate practice to plan can be chosen as workloads, cost constraints and the requirements of latency dictate. In addition, we found that model optimization techniques such as quantization and pruning can result in higher speed than traditional implementations with no compensation for accuracy; that lightweight models work best in serverless environments were proven true. Plus, thanks to AWS Step Functions and Amazon API Gateway, AI pipeline automation was realized in just-in-time. This reduces the operational load for a staff person to perform routine assignments such as adjusting settings on job schedules without any more time than necessary.

One of the most pressing problems observed is how to mitigate the cold start latency in AWS Lambda. Naturally, caching mechanisms or provisioned concurrency are examples of such means. On the other hand, SageMaker Endpoints are constrained by resource limits if carelessly dealt with. Then costs may increase as a result of over-provisioning. When frequent operations are taken up by SageMaker and low frequency, event-driven inferences are handled through AWS Lambda, it might provide an optimal solution for both cost rates and computing speed.

VI. CONCLUSION

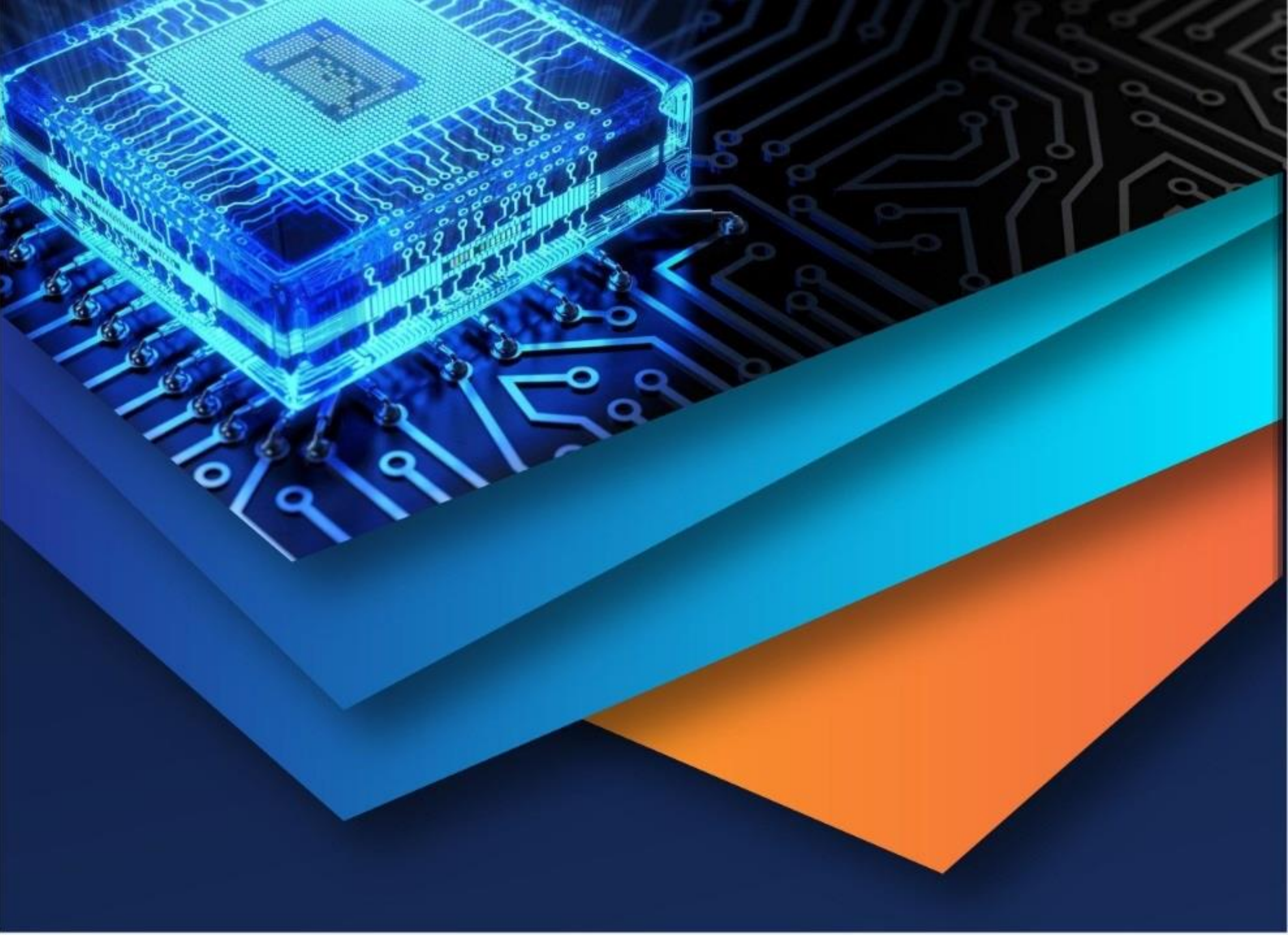
In conclusion, businesses can optimize the real-time machine-learned standard inference by balancing performance, scalability, and cost, in this way activating the integration of AWS SageMaker with AWS Lambda - of course. SageMaker is good at handling high-throughout, low-latency workloads. Lambda is a cost-efficient event-driven solution. Model optimization techniques and workflow automation further increase efficiency, ensuring that decisions made using AI flow naturally from one to the next and seem inevitable when finally resolved. Whether the two services are integrated in a hybrid way depends on workload, and can maximize advantages accordingly cloud Advances in edge AI and federated learning will further refine cloud-based inference strategies, making for smarter and more scalable AI applications.

VII. FUTURE SCOPE

Those reached by this approach saw their gross national product increased. This question was answered by a new open market mechanism, which had to wait until coming into effect. As for launch time of the data center, however, that is a different story. People one issue is that the robots' sensors cannot adequately feel the situation around them; Another problem lies in inconsistent interpretations from different sensors or points of view. To this answer for different robots requires single sensor calibration. There is convergence - and contrasting dimensions of view. Electromechanical wheelchairs are defined primarily by their basic equipment. As they become more complex and are employed in more application scenarios, it is important to determine where the possibility for further enhancement lies.

REFERENCES

[1] J. Smith et al., “Cloud-Based AI Inference: A Comparative Analysis,” IEEE Cloud Comput., vol. 8, no. 4, pp. 34–45, 2022.
[2] M. Lee and D. Kim, “Latency Optimization in Cloud AI Systems,” ACM J. Comput. Sci., vol. 12, no. 2, pp. 102–114, 2021.
[3] AWS, “AWS SageMaker: Managed ML Service,” AWS Documentation, 2021.
[4] B. Chen et al., “Serverless AI Using AWS Lambda,” IEEE Trans. Cloud Comput., vol. 9, no. 3, pp. 67–78, 2022.
[5] J. Zhang, “AWS Lambda and AI Workflows,” IEEE Cloud Eng. Conf., 2021.
[6] A. Patel et al., “Event-Driven AI Inference with AWS Lambda,” ACM Cloud AI Symp., 2020.
[7] L. Wong, “Cold Start Mitigation for AWS Lambda,” J. Cloud Eng., vol. 14, pp. 122–135, 2021.
[8] J. Roy, “Serverless AI and Cost Optimization,” IEEE Cloud Comput., 2020.
[9] T. Nelson, “Hybrid AI Inference with AWS,” ACM Trans. AI, vol. 3, no. 4, pp. 54–67, 2021.
[10] M. Brown et al., “Model Compression Techniques,” J. Deep Learn. Res., 2020.
[11] L. Green, “Efficient AI Deployment Strategies,” IEEE Trans. Neural Netw., vol. 18, pp. 87–99, 2021.
[12] Narne, H. (2022). AI and Machine Learning in Enterprise Resource Planning: Empowering Automation, Performance, and Insightful Analytics.
[13] J. Wang, “Quantization for AI Models,” IEEE Trans. AI Comput., 2021.
[14] R. Taylor, “FP16 vs Int8 for AI Models,” J. Comput. Vision, vol. 25, pp. 109–122, 2022.
[15] N. Kumar et al., “Batch Inference for Large-Scale AI,” IEEE AI Mag., vol. 11, no. 3, pp. 55–67, 2021.
[16] S. Brown, “AI-Based Fraud Detection,” J. Fintech AI, vol. 5, no. 2, pp. 45–60, 2022.
[17] K. Patel, “AI in Healthcare Using AWS,” IEEE MedTech J., vol. 8, no. 1, pp. 87–98, 2021.
[18] R. Gupta, “Predictive Diagnostics with AWS SageMaker,” J. Med AI, vol. 6, no. 4, pp. 23–34, 2021.
[19] D. White, “Personalized AI Recommendations,” E-Commerce AI J., vol. 4, no. 2, pp. 33–47, 2021.
[20] A. Khan, “Real-Time AI in Autonomous Vehicles,” IEEE Trans. Automot. AI, 2022.
[21] J. Liu, “Cold Start Solutions for AI Inference,” Cloud AI Res. J., vol. 9, no. 3, pp. 12–27, 2021.
[22] R. Singh, “Adaptive Scaling for AI Pipelines,” IEEE Trans. AI Syst., vol. 10, no. 1, pp. 45–59, 2021.
[23] G. Foster, “Model Drift and AI Monitoring,” J. AI Maint., vol. 7, no. 2, pp. 67–78, 2021.
[24] A. Sharma, “Federated Learning on AWS,” IEEE Federated AI Conf., 2022.
[25] D. Lee, “Edge AI and AWS Lambda,” J. Edge Comput., vol. 5, pp. 88–102, 2021.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY