International Journal of Computer Techniques – Volume 11 Issue 6, Nov - 2024

# Enriching Clickstream Analytics with Generative AI: A Scalable Serverless Architecture for Real-Time Session Intelligence

Vivek Venkatesan[1], Chakkaravarthy Arunachalam[2], and Rajesh Kumar Kanji[3]

[1]Independent Researcher, USA
[2]Independent Researcher, USA
[3]Independent Researcher, USA

Submitted: February 4, 2025

## Abstract

In large-scale digital ecosystems, understanding user behavior through clickstream data is essential for optimizing user experience, marketing strategy, and conversion funnels. Traditional approaches to session analysis often rely on static rules or manual reconstruction, which are difficult to scale and lack contextual nuance. In this paper, we present a scalable, serverless architecture that uses generative AI to extract session-level intent and behavioral summaries from Adobe Analytics clickstream data. The system integrates AWS Glue for session aggregation, Lambda for orchestration, and Claude via AWS Bedrock for large language model inference. Developed in a Fortune 500 financial company, the solution processes over 200,000 sessions daily, achieving 87% precision in manual QA validation, sub second inference latency and a 40% reduction in enrichment cost through prompt optimization. Our results demonstrate that AI-enriched session data significantly enhance analyst workflows by automating narrative generation, improving segmentation, and enabling intent-aware dashboards. We discuss key architectural components, prompt design strategies, and use cases in support detection, conversion analysis, and content engagement. This work establishes a blueprint for real-time behavioral intelligence using modern cloud-native and AI-driven technologies.

**Keywords:** Clickstream analytics, generative AI, user session enrichment, large language models, behavioral intelligence, serverless architecture, AWS Glue, Adobe Analytics, real-time inference, data engineering, cloud computing.

## 1 Introduction

Clickstream data serves as a foundational element in understanding user interactions across digital platforms. It captures detailed information about user actions—such as page views, clicks, searches, and form submissions—that can be used to optimize customer journeys, improve user experience, and drive business outcomes. Tools such as Adobe Analytics provide comprehensive hit-level data, yet analysts are often burdened with the manual task of stitching these events into coherent session narratives.

Traditional approaches for deriving session-level insights rely on rule-based segmentation or funnel logic. These methods are not only time-consuming and error-prone, but also fail to scale efficiently in high-volume environments. In addition, they offer limited adaptability to dynamic user behaviors and evolving web experiences.

Recent progress in natural language processing, particularly the emergence of large language models (LLMs), opens new opportunities to automate and enrich behavioral analytics. LLMs can infer user intent, summarize action sequences, and provide contextual understanding—capabilities that are especially valuable in the interpretation of largescale session data.

In this article, we introduce a cloud-native, serverless architecture that integrates LLMs into a production-scale Adobe Analytics pipeline. Our solution leverages AWS Glue for session aggregation, Lambda for orchestration, and Claude via AWS Bedrock for AI-driven inference. The system enriches session records with predicted intent and natural language summaries, enabling analysts to query user behavior more intuitively and act on insights more effectively.

### 1.1 Contributions

This paper makes the following key contributions:

- We design and deploy a fully serverless, cloud-native architecture that enriches Adobe Analytics clickstream sessions using generative AI.
- We introduce a prompt-based inference framework for extracting session intent and human-readable behavioral summaries from raw user interaction data.
- We evaluated the system at an enterprise scale, processing over 200,000 daily sessions with high enrichment accuracy and sub second inference latency.
- We demonstrate the analytical value of enriched sessions in multiple use cases, including support journey detection, conversion funnel analysis, and bounce behavior segmentation.

## 2 Related Work

Clickstream data has been extensively studied in the context of web usage mining and behavioral analytics. Traditional approaches often involve rule-based path analysis and segmentation techniques that depend on manually defined heuristics [2]. Although effective in controlled environments, these techniques struggle to scale or adapt to complex and dynamic user journeys.

To overcome these limitations, machine learning techniques have been employed to infer user intent and predict behavior. Jiang and Lee [1] proposed a semi-supervised approach that combines labeled and unlabeled session data to improve intent prediction. Similarly, Sharma and Dey [2] demonstrated how supervised learning can be applied to model the customer journey, although their approach still required significant manual feature engineering.

Recent advances in deep learning have further expanded the possibilities for interpreting sequential behavior in clickstream data. Shen et al. [4] utilized deep sequential models to predict visit dates on the next page and dwell durations, demonstrating improved accuracy in navigation forecasting. Jindal and Singh [3] explored deep learning methods to understand consumer navigation and decision-making behavior, focusing on the role of model-driven interpretation in e-Commerce contexts.

Large language models (LLMs) have opened new opportunities for extracting semantic meaning from unstructured or sequential data. Lakshmanaprabu et al. [5] reviewed the impact of LLMs on intelligent systems and emphasized their ability to replace traditional models in analytics workflows. However, most existing studies focus on offline or experimental settings rather than operationalized, production-grade deployments.

In contrast, our work integrates LLMs into a cloud-native, serverless architecture that supports real-time inference for session-level behavior enrichment. Unlike previous work, which relies on predefined labels or static models, our solution uses prompt-based reasoning to generate intent and narrative summaries dynamically, providing a scalable and interpretable layer of intelligence in live digital analytics environments.

# 3 System Architecture

The system is designed as a modular, cloud-native pipeline that ingests clickstream data from Adobe Analytics, aggregates user sessions, enriches them using a large language model, and makes the results available for analytical exploration. The entire architecture is built using AWS-managed services and is optimized for scalability, fault tolerance, and cost efficiency.

Figure 1 illustrates the high-level system architecture of the clickstream intent enrichment pipeline.

Figure 1. System Architecture for Clickstream Intent Enrichment Pipeline

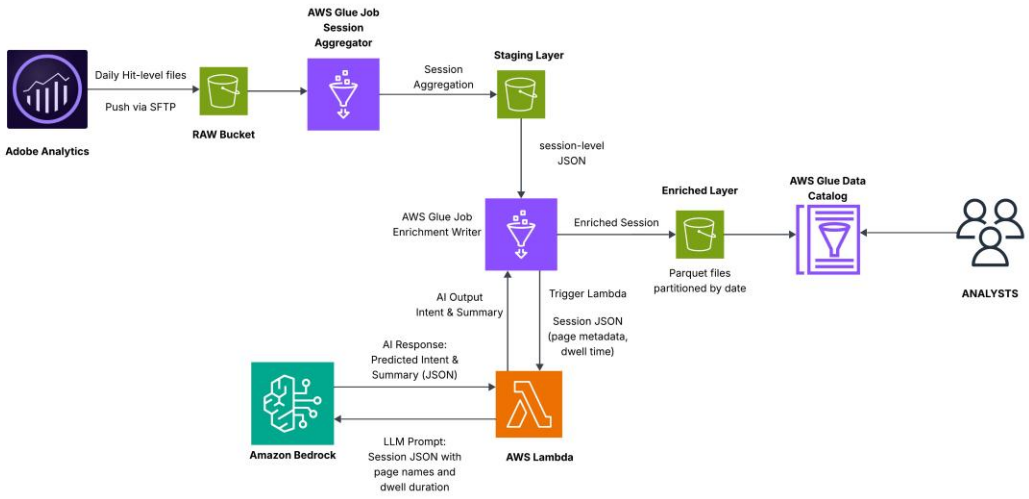


Figure 1: System Architecture for Clickstream Intent Enrichment Pipeline

The architecture consists of the following major components:

## 3.1 Data Ingestion

Hit-level clickstream data is generated from Adobe Analytics and delivered as flat files via SFTP. These files are ingested daily and stored in a raw S3 bucket in Parquet format.

### 3.2 Session Aggregation

An AWS Glue ETL job performs the task of grouping raw hits into session-level records. The job:

- Groups events by session id and sorts them chronologically.
- Computes metrics such as session duration, dwell time per page, first and last page.
- Produces a structured JSON representation of the page journey for each session.

The output is written to an intermediate staging layer in S3.

### 3.3 AI Enrichment Orchestration

A second AWS Glue job reads the session-level JSON records and initiates the enrichment process. For each session, the following steps are executed:

1. A Lambda function is triggered to handle inference requests.
2. The Lambda constructs a prompt containing session metadata (page names, dwell durations).
3. The prompt is sent to Amazon Bedrock, where the Claude model processes it and returns a predicted user intent and a narrative summary.

The AI-enriched output is attached to each session record.

### 3.4 Enriched Data Storage and Cataloging

The enriched sessions are written in an enriched layer based on S3 in parquet format. Data are partitioned by date for efficient querying. An AWS Glue Data Catalog table is maintained for seamless integration with Amazon Athena and BI tools.

### 3.5 Analyst Access and Exploration

Analysts can access the enriched session dataset using SQL via Athena or integrate the results into dashboards and business reports. This reduces the time required for manual segmentation and improves insight generation.

### 3.6 Scalability and Extensibility

The pipeline supports the parallel execution of glue jobs, stateless Lambda invocations, and asynchronous LLM calls. Additional capabilities such as persona tagging, funnel stage detection, or multi-intent classification can be integrated by extending the prompt structure or enriching the session schema.

# 4 Methodology

The clickstream intent enrichment pipeline consists of two primary processing stages: session aggregation and AI based enrichment. This section details the logic implemented within each AWS Glue job, the prompt engineering strategy used for large language model (LLM) inference, and the Lambda orchestration used to integrate with Amazon Bedrock.

## 4.1 Session Aggregation (Glue Job 1)

The first AWS Glue ETL job ingests raw hit-level data and produces structured session records. The key steps are the following.

- **Session Grouping:** Events are grouped by session id and ordered by hit timestamp.
- **Dwell Time Calculation:** For each page, the duration of dwell is calculated as the difference in time between consecutive events.
- **Metadata Derivation:** Features such as first page, last page, session duration, page sequence, and top engaged _page are extracted.
- **Session JSON Output:** A structured JSON object is constructed per session containing page names, URLs, and dwell times.

**Example Output Format:**

```
{
   "session_id": "s1",
   "pages": [
      {"page_name": "Homepage", "page_url": "/home", "dwell_duration": 68},
      {"page_name": "Loans", "page_url": "/loans", "dwell_duration": 122},
      {"page_name": "Apply Now", "page_url": "/apply", "dwell_duration": 10}
   ]
}
```

This output is written to the staging layer in S3, partitioned by data date.

## 4.2 AI-Based Session Enrichment (Glue Job 2 + Lambda)

The second AWS Glue job reads from the staging layer and invokes a Lambda function that performs LLM inference using Amazon Bedrock. The process consists of:

1. **Trigger Lambda:** For each session of JSON, a Lambda function is invoked asynchronously.
2. **Prompt Construction:** A structured prompt is constructed to include session metadata and dwell times in natural language format.

3. **LLM Inference:** The Lambda sends the prompt to Claude via Amazon Bedrock using the anthropic.claude-v2:1 model.
4. **Response Parsing:** The returned JSON containing the predicted intent and summary is appended to the session record.

**Sample Prompt**

Below is a user session containing the pages they visited and how long they stayed on each page.

Session ID: s1

1. Homepage – 68 seconds
2. Loans – 122 seconds
3. Apply Now – 10 seconds

Based on this information, please return the user's intent and a short session summary using the format: {"intent": "", "summary": ""}

**Expected Response:**

```
{
   "intent": "Apply Now",
                                "summary": "The user explored the homepage and loans page, then navigated to Apply Now. This suggests
}
```

The enriched session record is then written to an S3 output bucket (enriched layer) in Parquet format, ready for downstream analysis.

## 4.3 Observability and Logging

CloudWatch is used to monitor both Glue and Lambda execution. Detailed logs include:

- Prompt text and model response (Lambda)
- Inference duration and success status
- Glue job statistics (processed rows, partitions)

The design supports scalable enrichment through parallel partitioning, batch inference extension, and prompt optimization.

# 5 Evaluation and Results

We evaluated the system's performance, accuracy, and scalability using production-scale data from a Fortune 500 financial enterprise. The evaluation focused on three dimensions: enrichment latency, output quality, and cost efficiency.

## 5.1 Performance Benchmarks

The pipeline was benchmarked using over 200,000 user sessions processed daily. Table 1 summarizes the key performance metrics observed during deployment.

Table 1: System Performance Benchmarks

| Sessions | Glue Job Duration | Model Latency | Lambda Throughput | Enrichment Success Rate |
|---|---|---|---|---|
| 200,000 | ~22 minutes | ~1.2 seconds | 100 sessions/sec | 99.5% |

The serverless architecture enabled elastic scaling, with concurrent partitions in AWS Glue and asynchronous Lambda invocations maintaining low end-to-end latency.

## 5.2 Output Accuracy

To assess the quality of AI-generated outputs, a sample of 1,000 enriched sessions was manually reviewed by domain analysts. The predicted intent and summary were compared against human interpretations.

- Intent accuracy: 87%
- Summary clarity: 91% rated as business-actionable
- JSON structure validity: 100%

The prompt structure, the use of delimiters and the clear formatting contributed significantly to the consistency of the model response.

## 5.3 Cost Efficiency

The system incorporated prompt batching and prompt compression to reduce API call overhead. These optimizations led to an approximate 40% reduction in Bedrock inference costs, making the solution financially viable for high through put environments.

### 5.4 Observability Insights

CloudWatch metrics were used to monitor Glue runtime, Lambda error rates, and Bedrock latency. Logs helped identify prompt anomalies and session-level inference issues during pilot testing, enabling continuous improvement of the orchestration and enrichment logic.

## 6 Use Cases and Analyst Value

The AI-enriched session data set allowed digital analysts and product teams to derive actionable insights without relying on manual funnel reconstruction or static segmentation rules. This section outlines key use cases observed during internal deployment.

### 6.1 Intent-Based Segmentation

Analysts were able to filter sessions by inferred intent labels (e.g., Apply Now, Support Seeking, Retirement Planning) to investigate pre-conversion behaviors. This facilitated the identification of high-intent user paths and friction points prior to goal completion.

### 6.2 Support Journey Detection

Sessions marked with Support Seeking intent frequently contained repeated login attempts, error page visits, or help center interactions. These sessions were flagged for deeper UX investigation and operationalized into triage queues for customer support teams.

### 6.3 Bounce Analysis

Sessions labeled with Bounce intent were characterized by short dwell durations and single-page exits. By correlating bounce rates with landing page metadata, marketing teams could identify and redesign ineffective entry points.

### 6.4 Internal Search Summary

The use of search terms was automatically included in the session summaries. This provided context around what users were trying to find and whether the search journey led to deeper engagement or drop-off.

### 6.5 Most Engaging Page Detection

The top page per session dwell time was extracted as a proxy for content engagement. This metric allowed teams to identify which product pages retained user attention, even if conversions did not occur.

### 6.6 Narrative Intelligence for Stakeholders

Human-readable session summaries allowed product, UX and marketing teams to quickly interpret user behavior during A/B testing or campaign analysis, without writing SQL queries or combing through raw logs.

## 7 Future Work and Limitations

Although the current system delivers scalable and accurate session enrichment using generative AI, there are several avenues for enhancement and broader applicability.

### 7.1 Batch Prompting Optimization

Currently, inference is performed at the session level. A natural extension is to implement batch prompting, where multiple sessions are sent within a single LLM prompt. This approach could reduce API invocation overhead and lower cost per session, provided that prompt clarity and output consistency are preserved.

### 7.2 Persona and Journey Classification

By generating embeddings from session-level behavior or narrative summaries, clustering algorithms can be used to group users into behavioral personas (e.g., Research-Oriented, Conversion-Focused). These clusters could then be labeled using generative prompts, enabling higher-level audience segmentation.

### 7.3 Funnel Stage Inference

The inclusion of form events, login interactions, and process milestones could allow the model to infer user funnel stages (e.g., Form Abandoned, Application Completed). This would support conversion optimization and dropout analysis at a granular level.

### 7.4 Real-Time Streaming Integration

The current system operates in batch mode on daily data. A future version may use Amazon Kinesis or similar streaming infrastructure, along with Bedrock's asynchronous inference, to support near-real-time session tagging. This would allow live dashboards and alerts based on AI-inferred session states.

### 7.5 Embedding-Based Similarity and Visualization

With the use of vector embeddings (from summaries or full session paths), analysts could perform similarity-based retrieval or visualize clusters of similar behaviors. Tools such as OpenSearch, Pinecone, or vector-based dashboards could enable new exploratory workflows.

### 7.6 Richer AI Output

Prompt templates could be expanded to extract sentiment, urgency, or tone, and support multiintent categorization. This would provide a more nuanced understanding of user sessions beyond single-label intent classification.

### 7.7 Limitations

Despite its effectiveness, the current system has several limitations:

- **LLM Dependency:** The precision and consistency of enrichment outputs are directly related to the behavior of the Claude model, which may evolve over time or vary across versions.
- **Prompt Sensitivity:** Session summaries and intent predictions are sensitive to prompt formatting and token context, particularly for short or ambiguous sessions.
- **Latency Constraints:** While the system performs well in batch mode, scaling to real-time enrichment may require architectural modifications and cost-performance tradeoffs.
- **Domain Adaptability:** The current prompt design is optimized for financial services. Applying this solution to other industries may require new schemas, intent categories, and retraining QA baselines.
- **Lack of Ground Truth Labels:** Evaluation relies on manual QA due to the absence of labeled session-intent datasets. This introduces subjectivity and limits statistical benchmarking.

## 8 Conclusion

This paper presents a scalable serverless architecture for enriching Adobe Analytics clickstream data using generative AI. The system leverages AWS Glue for session aggregation, AWS Lambda for orchestration, and Amazon Bedrock (Claude) for large-language model inference. It transforms raw user interactions into enriched session records containing predicted intent and narrative summaries, thereby reducing the manual burden on analysts and accelerating insight generation.

Developed in a Fortune 500 company, the solution demonstrated strong scale performance, with high accuracy in inference, low latency, and substantial cost savings. The enriched session data enabled use cases across funnel analysis, support detection, bounce tracking, and behavioral segmentation.

By integrating LLMs into a cloud-native pipeline, this work highlights a new paradigm for behavioral analytics, one that combines the scale of big data with the interpretability of natural language generation. The approach establishes a blueprint for AI-augmented session intelligence, laying the groundwork for more adaptive, real-time, and user-centric digital analytics systems.

## Acknowledgments

The author would like to thank the platform engineering teams for maintaining the Adobe Analytics data pipelines and S3 data lake framework. The architectural design, the implementation of the glue pipeline, prompt engineering,

and Bedrock integration were solely led and executed by the author. Feedback from internal digital analysts was instrumental during the validation and refinement of the system.

## References

[1] Y. Jiang and D. Lee, "A semi-supervised learning approach to web user intention prediction," *IEEE Access*, vol. 6, pp. 24883–24894, 2018.

[2] A. Sharma and N. Dey, "Machine learning for customer journey analysis," *IEEE Intelligent Systems*, vol. 33, no. 4, pp. 40–47, 2018.

[3] R. Jindal and R. Singh, "Consumer navigation behavior in digital channels: An empirical study using deep learning," *Electronic Commerce Research*, vol. 21, pp. 429–452, 2021.

[4] H. Shen, R. Wang, and W. Song, "User behavior prediction based on deep sequential model," *Future Generation Computer Systems*, vol. 86, pp. 765–772, 2018.

[5] S. K. Lakshmanaprabu *et al.*, "Transforming intelligent systems with large language models: A review and future directions," *Multimedia Tools and Applications*, Springer, 2023.

## Author Biography

**Vivek Venkatesan** was born in India in 1988. He received his Bachelor of Engineering degree in Electrical and Electronics from Anna University, Chennai, India, in 2009.

He is currently a Lead Data Engineer at a Fortune 500 company, where he focuses on designing and implementing scalable data systems and cloud native pipelines. He has over 15 years of experience in data engineering, with contributions in the healthcare, insurance, and financial sectors. During the COVID-19 pandemic, he developed a health monitoring and exposure alerting system used by thousands of employees, significantly improving operational response and efficiency.

Vivek is a member of IEEE and actively serves as a reviewer for technical conferences and journals. His professional interests include data architecture, workflow automation, real-time analytics, and applying AI in enterprise environments.

**Chakkaravarthy Arunachalam** is the Head of Data Strategy, Analytics, and HRIS at a leading biotechnology firm, where he is responsible for building a scalable data foundation and advancing people technology systems. His primary focus is to serve as a trusted data advisor for executives, enabling evidence-based decision-making through data science and analytics.

Previously, he served as Senior Director of People Strategy and Analytics at Salesforce. Earlier in his career, Chakkaravarthy held engineering roles at Oracle and Visa, where he built large-scale data systems. His experience uniquely positions him to bridge deep technical expertise with strategic impact in the domain of workforce intelligence.

**Rajesh Kumar Kanji** is a highly experienced lead technical engineer with more than 16 years of experience in data integration architecture and cloud migration. He specializes in Informatica tools, IICS, and building scalable ETL/ELT solutions across platforms such as AWS, Azure, and Snowflake. He is an IEEE Senior Member and actively contributes as a judge for several industry awards, including the Globee Awards and Brandon Hall Group Awards. Rajesh also serves on the technical program committees of various international conferences and sits on the editorial boards of IJRTI and IJSDR, where he supports research publication and peer review.