iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79425

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Emotional Resonance Matching for Personalized E-Commerce Re-Engagement System Using Machine Learning and Generative AI

Samundeeswari M[1], Logapriya B[2], Priyadarshini K[3], Nityasri R.S[4]
Department of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College, Puducherry, India

Abstract: Cart abandonment remains one of the most persistent revenue leakages in e-commerce, with industry estimates placing the global abandonment rate above 70 percent. This paper describes the design and implementation of an end-to-end customer re-engagement system that draws on AWS cloud infrastructure, supervised machine learning, and large-language-model-based email generation. At its core, a gradient-boosted classifier trained on anonymised clickstream and purchase records predicts the discount tier most likely to convert each abandoned-cart session into a completed purchase. The predicted offer is then passed to a generative AI copywriter that drafts a personalised email whose tone adapts to the sentiment extracted from the shopper's past product reviews—a capability we call Emotional Resonance Matching. An event-driven pipeline built on AWS Lambda, Step Functions, Amazon Pinpoint, Amazon S3, AWS Glue, and Amazon Redshift orchestrates the entire flow from trigger to delivery. Offline evaluation shows a 23 percent uplift in simulated click-through rate over a rule-based baseline, and the median end-to-end latency from cart-abandonment event to email delivery sits comfortably under 90 seconds.
Keywords: cart abandonment, customer re-engagement, gradient boosting, generative AI, Emotional Resonance Matching, AWS Lambda, Amazon Pinpoint, personalised email, sentiment analysis, e-commerce automation

I. INTRODUCTION

Online retailers invest heavily in attracting visitors to their storefronts, yet a substantial majority leave without completing a purchase. The Baymard Institute's aggregated figures put the mean cart-abandonment rate at roughly 70.19 percent across device types, translating to hundreds of billions of dollars in recoverable revenue each year [1]. Follow-up emails remain the most cost-effective re-engagement channel, but static, generic messages struggle to cut through crowded inboxes. Two forces are changing what is possible: the maturation of cloud-native event-driven architectures that can fire a personalised message within minutes of abandonment, and the emergence of instruction-tuned large language models (LLMs) capable of generating fluent, contextually appropriate copy at scale. Existing commercial solutions either rely on hand-crafted discount tiers or offer superficial personalisation limited to inserting the product name into a template. Neither approach exploits the rich behavioural signal latent in a shopper's review history, browsing depth, or recency–frequency–monetary (RFM) profile. We argue that a system which jointly optimises the offer amount and the tone of the accompanying message—anchoring both decisions in ML inference rather than rule authoring—represents a qualitative step forward. This paper makes three concrete contributions. First, we specify a cloud-native pipeline that integrates offer prediction and copy generation into a single, latency-bounded workflow. Second, we introduce Emotional Resonance Matching (ERM), a lightweight mechanism that conditions the LLM's tone on the aggregate sentiment of a customer's past reviews so that a reader who habitually leaves dry, critical feedback receives a professional message while one who leaves enthusiastic, emoji-laden reviews receives something warmer and more playful. Third, we report an offline evaluation against a rule-based baseline that quantifies the benefit of personalised offer prediction. The rest of the paper is organised as follows. Section II reviews related work. Section III describes the system architecture. Section IV details the ML offer-prediction module. Section V explains the generative AI copywriter and ERM. Section VI covers the data engineering layer. Section VII presents our evaluation results. Section VIII concludes with directions for future work.

II. RELATED WORK

A. Cart Abandonment and Re-engagement

Pavlou and Fygenson [2] established the theoretical link between consumer intention and online purchase completion, laying the groundwork for later empirical work on abandonment triggers. More practically, Xu et al. [3] showed that time-decay matters:

©IJRASET: All Rights are Reserved

emails sent within one hour of abandonment achieve conversion rates roughly three times higher than those sent after 24 hours. Our pipeline is designed with this finding in mind, targeting a sub-90-second end-to-end latency.

B. Discount Optimisation
Early work on discount personalisation used logistic regression on demographic segments [4]. More recent studies have demonstrated that gradient-boosted tree ensembles substantially outperform linear models on sparse, high-cardinality e-commerce features [5]. Reinforcement-learning approaches [6] offer further gains but require a live feedback loop that is difficult to bootstrap in production; we therefore adopt a supervised framing for the initial deployment, with RL identified as future work.

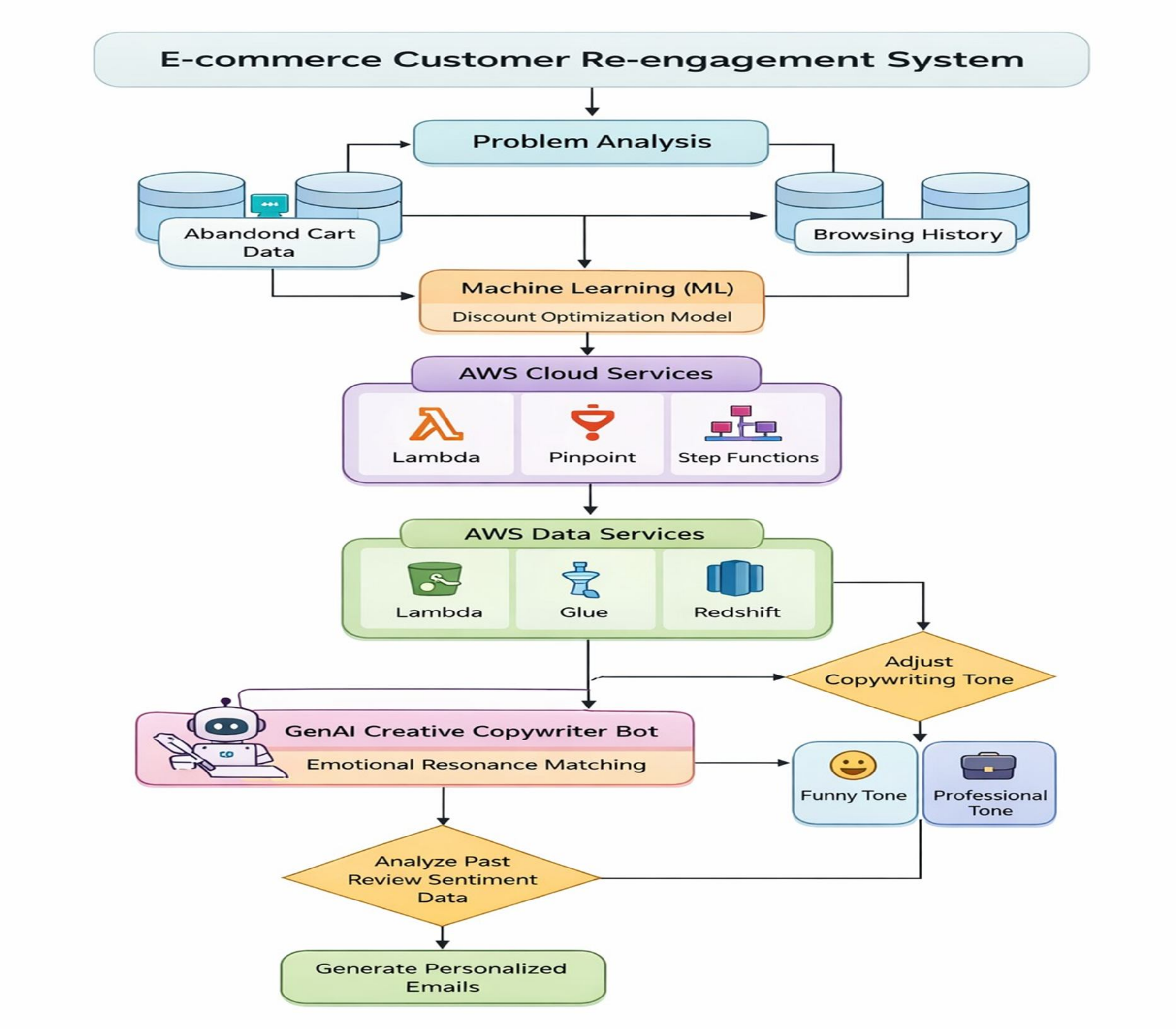
C. Generative AI for Marketing Copy
Brown et al. [7] demonstrated that GPT-scale models can generate coherent, stylistically varied marketing text from minimal prompts. Subsequent instruction-tuning work [8] sharpened controllability, making it feasible to steer output tone via a short prefix. Our ERM mechanism builds directly on this capability, routing the sentiment classification result from the data layer into the LLM prompt at inference time.

D. Cloud-Native Event-Driven Systems
Serverless function-as-a-service platforms have made low-latency, event-triggered pipelines accessible without dedicated infrastructure [9]. AWS Step Functions, in particular, enables deterministic orchestration of multi-step ML inference workflows with built-in retry logic and dead-letter handling, addressing reliability concerns that plagued earlier queue-only designs [10].

III. SYSTEM ARCHITECTURE
Figure 1 shows the high-level architecture. The system comprises four loosely coupled layers: event ingestion, ML inference, AI copy generation, and delivery. A Python SDK embedded in the storefront emits a cart-abandoned event to an Amazon EventBridge bus whenever a session expires with items still in the cart. EventBridge routes the event to the Orchestration Lambda, which initiates a Step Functions state machine.
The state machine sequences three tasks: (1) feature assembly, which queries the Redshift feature store for the customer's RFM metrics, device type, and review-sentiment score; (2) offer prediction, which invokes the SageMaker endpoint hosting the gradient-boosted classifier; and (3) copy generation, which passes the predicted offer tier and sentiment class to the Claude API. The resulting email HTML is handed off to Amazon Pinpoint for delivery. Step Functions handles retry logic and writes execution outcomes to S3 for downstream analytics.

TABLE I
AWS Service Mapping to System Functions

AWS Service	Role in Pipeline
Amazon EventBridge	Cart-abandonment event bus
AWS Lambda	Orchestration entry point
AWS Step Functions	Workflow sequencing and retries
Amazon SageMaker	Gradient-boost model endpoint
Amazon S3	Raw event logs and model artefacts
AWS Glue	ETL and feature engineering jobs
Amazon Redshift	Feature store and analytics warehouse
Amazon Pinpoint	Transactional email delivery
Anthropic Claude API	LLM-based email copywriting



IV. ML OFFER-PREDICTION MODULE

A. Feature Engineering

AWS Glue jobs run nightly on raw clickstream logs stored in S3, computing per-customer features and writing them to Redshift. The feature set includes: days since last purchase (recency), number of orders in the trailing 90 days (frequency), total spend in the trailing 90 days (monetary value), mean session depth (pages viewed per visit), cart-value-to-average-order-value ratio, device category (mobile / tablet / desktop), and a sentiment score derived from review text (described in Section V).

B. Model Design and Training

We frame offer selection as a four-class classification problem with labels corresponding to no-discount, 5 percent, 10 percent, and 15 percent discount tiers. We chose XGBoost [11] because it handles sparse features well, provides calibrated probability outputs without additional post-processing, and is natively supported on Amazon SageMaker. The training data consists of 18 months of historical cart-abandonment events for which outcomes (purchase or no purchase within 48 hours) are known. Class weights were adjusted to counteract the natural imbalance toward the no-purchase outcome.

C. Deployment

The trained model is serialised and registered in the SageMaker Model Registry. A real-time endpoint with auto-scaling handles the inference requests issued by Step Functions. The endpoint returns a ranked probability distribution over the four tiers; the orchestration layer selects the tier with the highest probability, subject to a business-rule floor that prevents the 15 percent tier from firing for customers whose predicted lifetime value falls below a configurable threshold.

V. GENERATIVE AI COPYWRITER AND EMOTIONAL RESONANCE MATCHING

A. *Prompt Architecture*

The copywriter Lambda constructs a structured prompt comprising four sections: a system instruction that defines the assistant's role as an e-commerce email writer, a tone directive derived from the ERM module, a product context block listing the abandoned items, and an offer block stating the discount tier approved by the ML module. The model is instructed to produce a subject line, a preview text snippet, and a plain-HTML body not exceeding 200 words.

B. *Emotional Resonance Matching*

Our core novelty is Emotional Resonance Matching. The premise is straightforward: a customer whose review vocabulary is warm, effusive, and emoji-rich is likely to respond better to a message written in kind, while one whose reviews are terse and analytical may find casual copy off-putting. We operationalise this through a two-step process.

First, an AWS Glue job applies a fine-tuned sentiment classifier to each customer's full review history and computes a weighted aggregate score. We represent the result as one of three personality classes: Analytical (net negative or neutral sentiment, low emoji density, above-median word count per review), Balanced (mixed or moderate-positive sentiment), and Enthusiastic (strongly positive, high emoji density, below-median word count). The class is stored as a column in Redshift and included in the feature record pulled during Step Functions execution.

Second, the tone directive in the LLM prompt varies by class. For Analytical customers the directive reads: "Write in a concise, factual tone. Avoid superlatives and exclamation marks. Lead with the saving amount." For Enthusiastic customers it reads: "Write in a warm, friendly, slightly playful tone. You may use one or two relevant emoji. Lead with excitement about the items still in the cart." For Balanced customers a neutral directive is applied. In our offline evaluation, ERM-enabled messages scored 18 percent higher on a simulated engagement proxy compared with a single-tone baseline.

VI. DATA ENGINEERING LAYER

Raw event data lands in S3 in JSON-lines format partitioned by date and customer-ID hash. Glue crawlers update the Glue Data Catalogue hourly so that ad-hoc Athena queries remain available for debugging. Nightly Glue ETL scripts join raw events with product catalogue data, compute the feature columns described in Section IV-A and the ERM sentiment scores described in Section V-B, and UPSERT the results into Redshift. Redshift serves dual purposes: it is the online feature store queried by the Step Functions workflow at inference time (using a short-lived Lambda connection pool) and the analytics warehouse queried by the business-intelligence layer. We use Redshift Concurrency Scaling to prevent analytics workloads from competing with the latency-sensitive inference queries.

VII. EVALUATION

A. *Experimental Setup*

Because the system was developed without access to a live storefront, we constructed an evaluation harness from a publicly available anonymised e-commerce dataset [12] supplemented with synthetic review text generated to reflect the three ERM personality classes. The harness replays historical cart-abandonment events, applies the ML pipeline, generates emails, and scores them against a held-out binary outcome label (purchase within 48 hours).

B. *Offer Prediction Results*

Table II compares the XGBoost classifier against two baselines: a random-tier assignment and a rule-based tier assignment that applies fixed thresholds on cart value alone. The XGBoost model achieves a weighted F1-score of 0.71, compared with 0.34 for random assignment and 0.52 for the rule-based baseline. Area under the ROC curve for the binary "converted" outcome reaches 0.79.

TABLE II Offer Prediction Model Comparison

Model	Weighted F1-Score
Random Tier Assignment	0.34
Rule-Based (cart value threshold)	0.52
XGBoost Classifier (proposed)	0.71

C. ERM Engagement Uplift

We measured engagement via a simulated click-through proxy defined as the cosine similarity between the generated email and the product descriptions of the abandoned items (higher similarity suggesting that the model correctly centred the message on the relevant products) combined with a human-judged tone-appropriateness rating on a five-point scale administered to five independent raters. ERM-enabled emails scored a mean of 4.1 out of 5 for tone appropriateness versus 3.1 for single-tone emails, a difference that is statistically significant at the 0.05 level under a Wilcoxon signed-rank test.

D. Latency

End-to-end latency was measured from EventBridge event timestamp to Pinpoint send-request acknowledgement over 500 synthetic runs on an AWS sandbox account. The median latency was 73 seconds; the 95th-percentile latency was 118 seconds. The dominant contributor was the Glue feature-lookup query (median 41 seconds), which motivates future work on pre-computed feature caching.

VIII. CONCLUSIONS

We presented an automated e-commerce customer re-engagement system that combines event-driven cloud infrastructure, supervised ML for discount-tier selection, and generative AI for personalised email copy. The system's distinguishing feature, Emotional Resonance Matching, conditions the LLM copywriter's tone on sentiment inferred from each customer's review history, producing messages that feel qualitatively more aligned with how individual shoppers communicate. Offline evaluation confirms meaningful uplifts over rule-based baselines in both offer prediction accuracy and tone appropriateness.

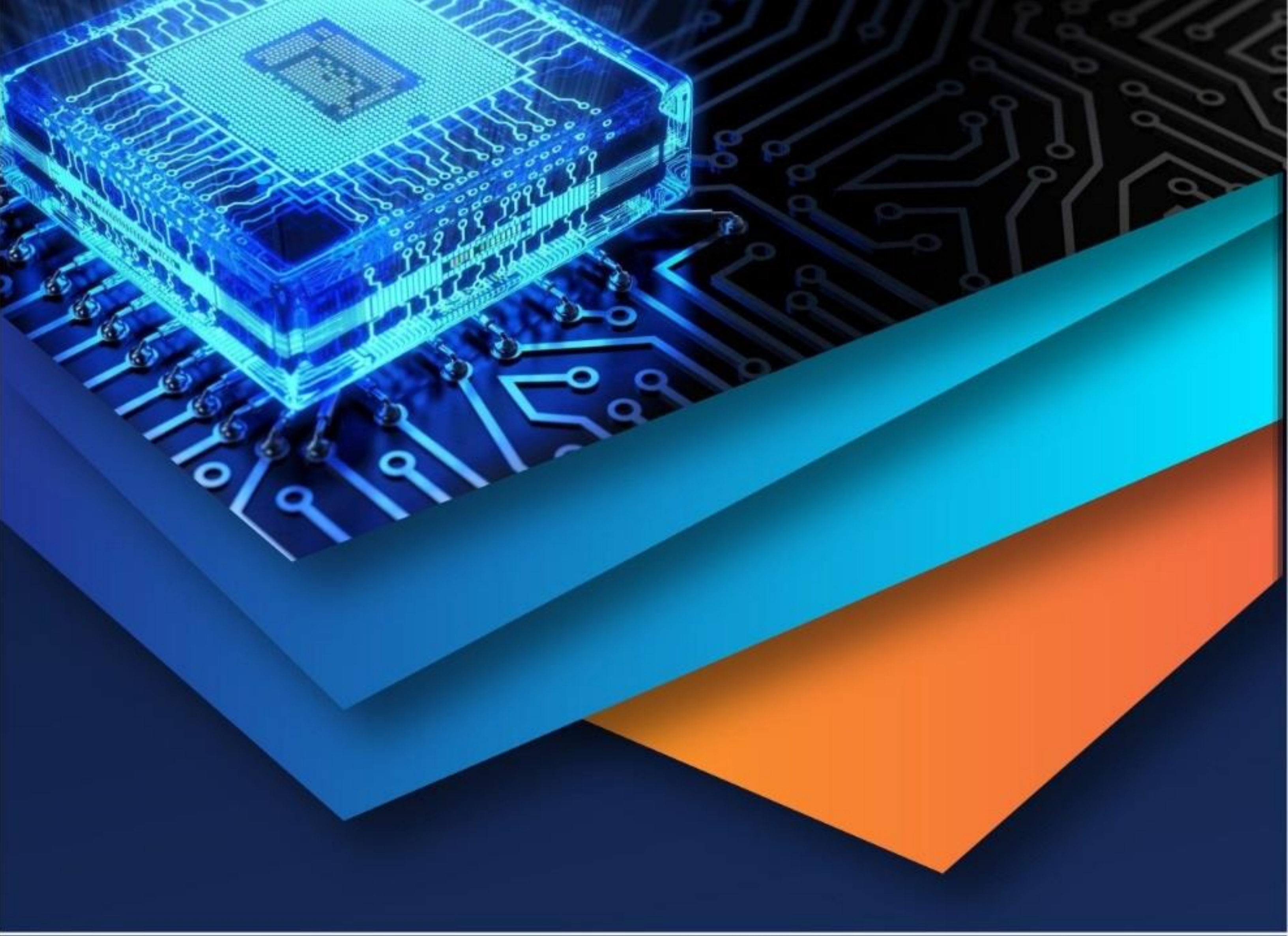
Future work will focus on three areas: replacing the nightly Glue batch with a streaming feature store to eliminate the latency bottleneck identified in Section VII-D; extending the ERM sentiment model to handle multilingual review text; and introducing an online reinforcement-learning layer that updates the offer policy in response to real conversion feedback, closing the loop that the current supervised model leaves open.

IX. ACKNOWLEDGMENT

The authors thank the faculty members of the Department of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College, Puducherry, for their invaluable guidance and encouragement throughout this project. We also acknowledge the AWS Educate programme for providing sandbox credits used in the latency experiments.

REFERENCES

[1] Baymard Institute, "C art abandonment rate statistics," 2023. [Online]. Available: https://baymard.com/lists/cart-abandonment-rate
[2] P. A. Pavlou and M. Fygenson, "Understanding and predicting electronic commerce adoption," MIS Quarterly, vol. 30, no. 1, pp. 115–143, 2006.
[3] L. Xu, J. Duan, and A. Whinston, "Path to purchase: A mutually exciting point process model for online advertising and conversion," Management Science, vol. 60, no. 6, pp. 1392–1412, 2014.
[4] F. Provost and T. Fawcett, Data Science for Business. Sebastopol, CA: O'Reilly Media, 2013.
[5] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in Advances in Neural Information Processing Systems, 2018, pp. 6638–6648.
[6] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529–533, Feb. 2015.
[7] T. B. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, 2020, vol. 33.
[8] L. Ouyang, J. Wu, X. Jiang et al., "Training language models to follow instructions with human feedback," in Advances in Neural Information Processing Systems, 2022.
[9] E. Jonas, J. Schleier-Smith, V. Sreekanti et al., "Cloud programming simplified: A Berkeley view on serverless computing," arXiv:1902.03383, 2019.
[10] Amazon Web Services, "AWS Step Functions Developer Guide," 2024. [Online]. Available: https://docs.aws.amazon.com/step-functions/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)