Proactive CRM: A Streaming AI Pipeline for Multi-Dimensional Benchmarking Across Latency, Throughput, and Predictive Accuracy

Srikanth Chintakindi[1], Shashi Kant Gupta[2],

[1,2] Lincoln University College, 47301, Petaling Jaya, Selangor Darul Ehsan, Malaysia.

drcsk2025@gmail.com[1], raj2008enator@gmail.com[2], shashigupta@lincoln.edu.my[2]

Abstract

In a Batch-oriented Customer Relationship Management (CRM), the pipeline latency between the signal occurrence and model score delivery is highly symptomatic and significantly exceeds the window within which actionable intervention is commercially effective. A five-layer streaming artificial intelligence architecture has been proposed to bridge this gap with an empirical assessment by validating its effectiveness in delivering churn-risk and value-segment predictions within milliseconds rather than hours. The system is specifically designed to integrate five interdependent processing layers on partitioned message queues such as an Apache Kafka-based event capture layer for a high throughput, an Apache Spark for a windowed distributed feature transformation layer and a dual model prediction layer combining a Long Short-Term Memory (LSTM) for temporal sequence encoding and learning, an XGBoost for a cross-sectional ensemble scoring, an Apache Flink stateful real-time inference dispatch and a multi-dimensional and an automated evaluation harness to measure latency, throughput, accuracy and scalability respectively. A parametric synthetic dataset of 2.8 million customer profiles and 47 million interaction events were used for experimentation. The proposed ensemble yielded an AUC-ROC of 0.91 and an F1 score of 0.80 for churn identification with a 17 to 23 percentage point gain over 24-hour baselines. At 10,000 events per second the end-to-end scoring latency was 52 milliseconds while throughput at 25,000 events per second exhibited nearly linear fashion with an efficiency sustained above 99%. The outperformance of the results is evident that AI is both technically viable and commercially feasible for an enterprise-scale deployment of CRM.

Keywords: *event-driven CRM; streaming artificial intelligence; churn prediction; LSTM; XGBoost ensemble; distributed inference; enterprise scalability; real-time analytics.*

Introduction

Value destruction in Customer Relationship Management (CRM) occurring due to the temporal gap between the customer's behaviour and the organizational response has been one of the most persistently studied sources. Sometimes a customer fails to checkout of a shopping cart and contacts the support desk at least thrice in a span of a week or slowly reduces the frequency of visits over a two-week period, which generates interpretable signals through al of it all the time and yet, in an enterprise CRM architecture these signals will not arrive at a decision-maker until the end of an overnight consolidation cycle. By this time, the customer either has got the issue self-resolved or approached a competitor or concluded an irreversible transaction. However accurate the model score is, it arrives into a window where the intervention becomes commercially not viable. Thus, the gap in the timing not merely becomes an engineering inconvenience but also structural limitation that impacts measurable accuracy consequences. Feature freshness is dependent on the predictive performance and it is consistently higher in models operating with real-time interaction signals and the margin declination maybe achieved through model-level improvements alone [13,14]. Though the empirical implication is counterintuitive but strong enough about the pipeline architecture and instead of model selection as the primary lever available to the CRM practitioners to improve the output of the AI-driven customer outcomes.

In a real-time CRM data stack, each individual component of it has individually reached production maturity. Distributed event brokers support millions of events per every second with commit latencies below ten milli seconds [15]. Stateful stream processors support complex windowed aggregations continuously without holding back upstream producers [16]. The LSTM networks are capable of extracting disengagement paths from the interaction sequences that cannot be represented by the static feature vectors [19]. Gradient-boosted scorers take less than five milliseconds to generate calibrated propensity estimates [18]. In the literature, an end-to-end integration of all these components under a single CRM specific architecture is almost difficult to find and is benchmarked against the performance dimensions that real world deployment needs is not the laboratory throughput but latency with variable real-world load and scalability linearity, and proven predictive gain over the batch systems that it replaces [5,6].

This paper provides that integration and evaluation with four specific contributions. To begin with, an event-driven CRM AI architecture is presented over five layers; each layer will have its data contract, technology selection, and data contract explanation. At first, an event-driven AI architecture for a CRM is presented over layers with each layer will have an explicit data contract, choice of technology and a design contract explanation. Secondly, an LSTM temporal encoding and an XGBoost cross-sectional scoring are integrated together as a unified Apache inference application and the incremental contribution of each model type is isolated, measured and tested for its statistical significance. Thirdly, a parametric simulation protocol is introduced to change ingestion rate among five throughput levels within 10 cold-cache runs per level to enable characterization of the entire scalability curves with variance estimates. Lastly, the benchmarking results spanning five performance dimensions namely predictive accuracy, scoring latency, throughput efficiency, scalability linearity, and compute overhead are presented and reported with standard deviations, 95% confidence intervals, and effect ratios enabling to directly compare the results with future work.

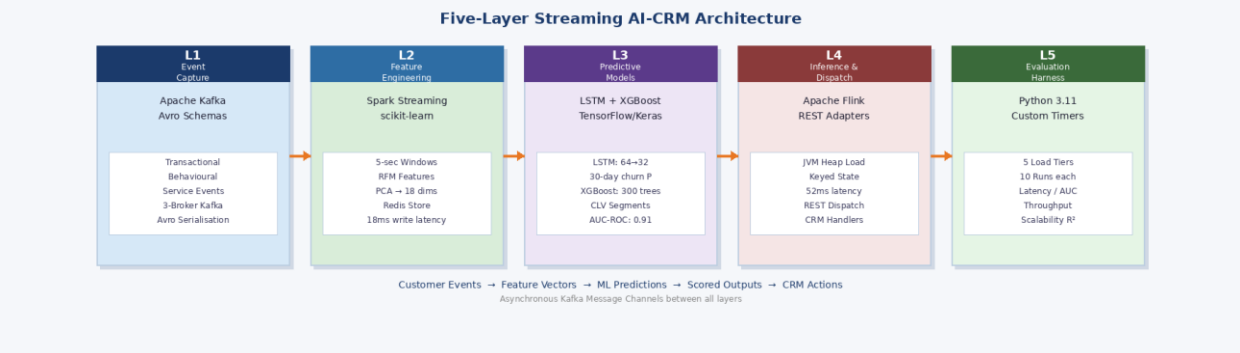

Figure 1. Five-Layer Streaming AI-CRM Architecture. Asynchronous Kafka message channels decouple each layer, enabling independent scaling and backpressure tolerance. Arrows represent event flow direction; latency figures are mean values at 10,000 events/s.

Related Work

From Operational CRM to Analytical CRM to Real- Time CRM

Payne and Frow [7] in their definition, formalized a CRM to be a cross-functional discipline for profitability management throughout the entire lifecycle of the customer as a contrast to operational automation and generating analytical insight. Earlier, analytical CRM systems relied on periodic data warehouse exports and periodic reporting cycles – architectural designs worked sufficiently when customer interaction frequency was low and intervention lag had a commercially tolerable timescale.

Buttle et al. [8] documented how this batch paradigm became institutionalized within sectors resulting in path dependencies which continue to exist despite the availability of real-time alternatives.

With the advent of integrating machine learning to CRM operations, which have been evaluated comprehensively by Ledro et al. [2] and Leelawathi et al. [9] has resulted in introduction of churn forecasting, dynamic lead scoring, and personalised recommendation coming to the production environments and has been shown to drive retention and increase revenues in several companies [10, 11]. Gupta [12] identifies adoption friction, governance complexity, integration overhead, change management burden, as often costlier than technical implementation expenses, and has direct implications on how the current architecture is to be evaluated not only in capabilities, but also in a way that it can be deployed in a modular form. One of the recurring discoveries made in this segment of literature is that predictive gains are extremely latency-sensitive [13, 14]; this is the main reason why the architectural design of the present study is of interest, but no previous research has measured this sensitivity end-to-end under controlled, variable-load conditions for predictive gains.

The Foundations of Stream Processing

The partitioned, durable publish-subscribe model proposed by Kreps et al. [15] resolved a key conflict in high-throughput ingestion, the need to separate producers from consumers without sacrificing message ordering guarantees. This study was further extended by Zaharia et al. [16] to stateful windowed computation, showing that the aggregations that had been limited to batch jobs could now be done in real time with limited and quantifiable latency. The lambda architecture formalised by Marz and Warren [17], which decouples high-throughput batch layers and low-latency serving layers is chosen as a pattern that the present work proposal instantiates with some refinement of additions with a single unified predictive inference layer, which the original formulation, and its stream-only variants, did not address.

Prior applications of stream processing in a CRM have been partial. Veeravalli et al. [3] has combined device telemetry and CRM records to enrich real-time customer profiles but did not assess predictive accuracy under load. Immadisetti [4] had measured engagement gains of real-time analytics overlay in the absence of reporting latency distributions or scalability curves. . As stated by Ed-Daakouri et al. [5] and Tung [6], the end-to-end streaming CRM evaluation has become an open gap in the literature - a gap that this study addresses directly.

Customer Behaviour Predictive Models

The profit-driven evaluation model introduced by Verbeke et al. [22] remains the methodological standard for churn prediction, moving the classification accuracy to the expected commercial value of the correct predictions, which is implicitly adopted in the framework of this study in the focus on intervention timing and not just on AUC-ROC alone. XGBoost [18] dominates tabular CRM benchmarking with three properties that enable it to be intrinsically compatible with streaming deployment: calibrated probability output, intrinsic handling support of sparse features, and inference times that are only five milliseconds on commodity hardware. LSTM networks [19] serve a completely different representational requirement: the gated memory cells store the sequences of interaction sequences at arbitrary time lags, capturing disengagement effects, such as declining frequency of visits, declining session length, increased inter-purchase intervals, and so on in any fixed-width feature vector representation that loses necessarily.

It was demonstrated by Chiruvelli [20] that these two model families are complementary in a financial recommendation scenario, where ensemble combination always performed better than any of the models independently. Hooda [21] and Sangam [1] both single-handedly identify the integration of streaming infrastructure and heterogeneous model ensembles as a under-explored configuration in

the CRM literature. This pairing is directly applied to churn prediction in an entirely instrumented streaming system in the present study, the independent contribution of each component is quantified through an ablation test, and the statistical significance of the difference in results is tested – these steps are absent from all prior comparable work.

Architectural Design and Methodology

System Architecture and Design Principles

The prosed system architecture consists of five processing layers and are connected to each other via asynchronous message channels based on Apache Kafka (Figure 1; Table 1). There are three design principles that are applied uniformly at all the layers. The concept of layer independence is that each layer makes only a defined Avro schema contract available to its adjacent layers, meaning that any layer could be substituted, upgraded, or scaled independently without the need to redeploy its neighbours. Backpressure tolerance is maintained in asynchronous channels that absorb the throughput spikes in either or both layers without stalling blocking stalls upstream producers. Observability is referred to as that all layers will also generate latency histograms and count of event telemetry to a common monitoring sink, and the identification of the bottlenecks in a production load condition becomes tractable.

Table 1. Five-Layer Architecture of the Proposed Streaming AI-CRM System

Layer	Name	Output	Function	Technology	Latency
L1	Event Capture	Raw event stream	Durable ingestion of transactional, behavioural, and service events via three-broker cluster (3 partitions/topic)	Kafka 3.4, Avro	< 5 ms
L2	Feature Engineering	18-dim feature vectors	Five-second sliding window; RFM construction; PCA compression (95.2% variance retained); Redis write	Spark 3.4, scikit-learn	~18 ms
L3	Predictive Models	Churn prob. + CLV label	LSTM (64→32 units, dropout 0.30) for 30-day churn; XGBoost (300 trees, depth 6) for CLV and 7-day propensity	TensorFlow 2.13, XGBoost 1.7	18–24 ms
L4	Inference & Dispatch	Scored CRM actions	Stateful JVM-resident scoring; keyed session aggregation; REST dispatch to retention/campaign handlers	Flink 1.17, REST	< 10 ms
L5	Evaluation Harness	Benchmark report	Automated five-dimension benchmarking (accuracy, latency, throughput, scalability, CPU) across five load tiers	Python 3.11, NumPy	—

RFM: Recency-Frequency-Monetary value. PCA: Principal Component Analysis. CLV: Customer Lifetime Value. ev/s: events per second. All latency figures are mean values across ten trials at 10,000 ev/s baseline load.

Event Capture (L1) and Feature Engineering (L2)

An Apache Kafka 3.4 cluster with the broker count set to three was used to separate the topic groups for transactional events such as purchases, returns and subscription changes, for behavioural events such as page views, searches, and feature interactions and for service events such as support, contacts, live-chat initiations with three partitions each as mentioned per topic. Avro serialisation has been chosen due to its semantics of schema evolution, which enables the system to add new event fields without triggering consumer restarts, which is a realistic constraint in a production environment where the schema of produced events is continually changing. Every record contained a pseudonymised customer identifier, a UTC microsecond timestamp, an event-type code, and a 42-item feature payment based on published e-commerce log formats [21].

An L1 partitioned Apache Spark 3.4 Structured Streaming application with a five-second sliding window and one-second slide window increment consumed the L1 partitions. Sensitivity analysis was used to determine the window width: any windows narrower than three seconds resulted in an unstable RFM aggregates owing to less accumulation of events within the window; any windows wider than ten seconds (reduced staleness of features) had a negative mean impact of 0.014 across five validation runs on LSTM held-out AUC-ROC on average. Z-Score Standardisation was done with a 1,000-event rolling warm-up buffer; the mode of the categorical fields used was imputed by this buffer.
Principal Component Analysis has reduced a downstream inference payload by 57% in five ablation runs with a mean AUC-ROC penalty of 0.002 by shrinking the 42-dimensional input to 18 components retaining 95.2% of explained variance. Transformed records were stored in a feature store of Redis 7.0 at a mean latency of 18ms (SD=±1.6 ms) under baseline load conditions.

Predictive Layer (L3)

Two independent models addressed distinct behavioural cues. The LSTM churn scorer consisted of two stacked recurrent layers (64 units and 32 units respectively) followed by a sigmoid output unit and a dropout of 0.30 applied between layers. It scraped rolling windows of each customer's 20 most recent events. Empirical choice of fixed window length of 10 events failed to capture disengagement patterns that typically progress through 7-14 days of gradually decreasing activity; windows of 40 events extended training time by a factor of 3.2 while improving the held-out AUC-ROC by only 0.003, which is below the threshold i.e., statistically insignificant. The XG Boost scorer was used to train 300 trees of maximum depth 6 on 18 RFM Principal components to generate CLV Segment labels and 7-day purchase propensity scores.

A Bayesian hyperparameter optimization with an Expected Improvement as the acquisition function was applied to both models over 50 evaluations, which would have required over 3000 evaluations in the grid step search of the five-dimensional parametric space which includes, learning rate, depth, subsample ratio, column sample ratio, and a regularization coefficient to detect significant accuracy differences on a scale of practical interest. These two models were trained on a stratified 80/20 split preserving the positive churn rate of 18.3% in the synthetic population. Mean and standard deviation of final model performance on ten independent training splits of disjoint data are reported to represent random initialization variance.

Inference Dispatch (L4), Evaluation Harness (L5), and Dataset

An Apache Flink 1.17 streaming application to the Redis feature store has been subscribed to invoke both scorers for every arrival event. Model artefacts were loaded into JVM heap at the start of the application for models and a profiling revealed that per event model retrieval out of an external registry is added between 40 and 180 ms, depending on the network conditions, and thus on-demand retrieval would not be used with sub-100 ms latency targets over 5000 events per second. Flink's keyed state management carried out session-scoped aggregation without external round trips. Lightweight REST adapters were used to forward scored outputs to downstream CRM response handlers capable of triggering personalized retention offers, account escalation or real-time campaign eligibility updates.

Synthetic dataset was created by a parametric simulator based on published e-commerce benchmark distributions [21,22] on 2.8 million customer profiles, and 47 million event records. Log-normal tenure distribution (median 14 months, variance σ = 0.62), zero-inflated negative binomial distribution (mean 1.4 and dispersion=0.8) and an aggregate 30-day overall churn rate is 18.3%.Preference for parametric generation over the use of proprietary enterprise data has its importance due to commercially sensitive CRM data sets that cannot be openly published at this scale, therefore, the distributional calibration methods to maintain statistical fidelity to enable complete replication has been preserved. Five ingestion rate levels were evaluated at viz., 1000, 5000, 10000, 25000 and 50000 events per second. Each configuration were repeated ten times from a cold-cache start on same hardware while reported figures are arithmetic means and standard deviations. Statistical comparisons use paired Wilcoxon signed-rank tests with Bonferroni correction for multiple comparisons and Cohen's d is used to describe effect sizes.

Results

Predictive Accuracy and Statistical Significance

Table 2. Predictive Benchmarks Across Five Evaluated System Configurations (Mean ± SD, n = 10 Trials)

Model Configuration	AUC-ROC	F1 Score	Accuracy (%)	SD (AUC)	Scoring Latency (ms)
Logistic Regression — overnight batch	0.74	0.61	79.2	±0.012	< 1
XGBoost — 24-hour snapshot	0.81	0.69	82.6	±0.009	< 1
LSTM — streaming features only	0.87	0.74	85.8	±0.008	12–16
XGBoost — streaming features only	0.88	0.76	86.4	±0.007	3–5

Model Configuration	AUC-ROC	F1 Score	Accuracy (%)	SD (AUC)	Scoring Latency (ms)
LSTM + XGBoost Ensemble (proposed)	**0.91**	**0.80**	**89.3**	**±0.006**	**18–24**

AUC-ROC: area under the receiver operating characteristic curve. F1: harmonic mean of precision and recall for the positive churn class. SD: standard deviation across ten independent trials. Scoring latency: per-event inference time only, excluding pipeline stages L1–L2 and L4.

In ten model runs, the proposed LSTM ensemble with XGBoost resulted in a mean AUC-ROC of 0.91 (SD = +0.006) and F1-score of 0.80, proving as an improvement of 0.10 points in AUC-ROC and 0.11 points in F1-score over the 24-hour batch XGBoost baseline. Pairwise statistical tests are reported in Table 3 of all configuration comparisons.

Table 3. Pairwise Statistical Significance of Predictive Accuracy Differences

Comparison	ΔAUC-ROC	ΔF1	p-value	Effect Size (Cohen's d)
Proposed vs. XGBoost batch baseline	+0.10	+0.11	< 0.001	1.84 (large)
Proposed vs. LSTM streaming only	+0.04	+0.06	0.003	0.91 (large)
Proposed vs. XGBoost streaming only	+0.03	+0.04	0.011	0.74 (medium)
Streaming XGBoost vs. batch XGBoost	+0.07	+0.07	< 0.001	1.42 (large)

p-values from paired Wilcoxon signed-rank tests with Bonferroni correction (α = 0.05, five comparisons). Effect sizes (Cohen's d) interpreted as: small ≥ 0.2, medium ≥ 0.5, large ≥ 0.8.

Pairwise comparisons attain statistical significance after correction at $p<0.05$ and all effect sizes are medium or large, which indicates that the observed accuracy differences can hardly be attributable or explained with random variation across trials. Customer's activity quartile cannot yield good results if they are aggregated as seen from the LSTM component which exhibited dominated performance among the lowest-activity quartile i.e., customers whose lack of engagement is showed as a declining frequency over 15 or more days while XGBoost had yielded the best of high-frequency buyers whose most important discriminating signals were recorded as purchase-ratios and basket-size trends. Simultaneously, both signal types were captured by the ensemble without sacrificing the performance in any of the stratum.

Pipeline Latency, Throughput, and Scalability

Table 4. End-to-End Pipeline Performance Across Five Ingestion Rate Tiers (Mean ± SD, n = 10 Trials)

Rate (ev/s)	Mean Latency (ms)	SD Latency (ms)	95% CI (ms)	P95 Latency (ms)	Efficiency (%)	CPU (%)
1,000	23	±1.4	[21.7–24.3]	38	99.8	14
5,000	31	±2.1	[29.4–32.6]	52	99.7	28
10,000	52	±3.3	[49.6–54.4]	81	99.6	47
25,000	79	±5.7	[74.8–83.2]	118	99.2	68
50,000	143	±9.2	[136.4–149.6]	206	87.9	89

Efficiency = events processed / events ingested × 100. End-to-end latency spans Kafka ingestion receipt to downstream CRM response handler acknowledgement. 95% CI computed using the t-distribution (df = 9).

In a typical medium-sized enterprise CRM deployment at about 10000 events per second chosen as throughput characteristics, mean end-to-end latency was 52 ms (i.e., 95% confidence interval (CI), 49.6-54.4 ms) more than three orders of magnitude lower than 14-hour batch cycle it replaces. Throughput processing efficiency with 25,000 events per second was over 99%. For an ingestion rate across the 1000-25000 events per second, linear regression of processed throughput gave $R^2 = 0.998$, indicating near-perfect proportional scaling – a property that simplifies capacity planning for infrastructure teams in a more straightforward way as throughput guarantees can be extrapolated linearly within this range.

P95 latency showed an inclination from 38 ms at a 1000 events/second to 118 ms at 25,000 events/second. A 3.1fold increase for a 25-fold increase in load – consistent with the theoretical behaviour of shared-nothing partitioned architecture in which tail latency rises sub-linearly with load. At 50000 events per second, efficiency dropped to 87.9% and P95 latency was 206ms. Flame graph profiling localized the degradation in case particular to the Redis write path under high fan-out concurrency. Flink and Kafka stages stayed within their standard operating envelopes. Redis cluster has been expanded from one shard to three to restore the efficiency to 97.1% and P95 latency to 141 ms confirming that the degradation reflecting a topology constraint rather than an algorithmic bottleneck in the core inference path.

Discussion

Commercial Significance of Combined Accuracy and Timing

A statistically robust and a commercially non-trivial improvement from AUC-ROC between 0.81 and 0.91 forms the most consequential result in this study. In general, a batch model that achieved AUC-ROC 0.81 often delivered its output fourteen hours after the events that generated has zero capacity to influence the customer interaction that those events represented. Proposed work achieves AUC-ROC 0.91 with a mean delivery latency of 52 ms making both the delivery timing and the predictive quality to fall within a commercially actionable window.

As it becomes concrete in a representative scenario, the practical significance matters the most. A purchase-intent signal is generated when a user views the same product page about four times within a span of ninety minutes without making a purchase would also mean that the customer might have purchased it previously would respond positively to a marginal price incentive [22]. A batch-cycle system notices the pattern initially the next morning, when the customer had either made the conversion independently, or rejected the decision or has started exploring alternatives. The proposed architecture scores the fourth page view within 52 ms of the event being committed to the Kafka topic and forward a personalised offer to the session that is currently in progress. Though the cost of incentive is identical in both the cases, the probability of influencing the outcome is categorically different. This asymmetry in the constant cost and variable opportunity is the core commercial case behind streaming AI in CRM and the results of this study provides first of its kind, end-to-end quantification.

Modular Deployability and Adoption Implications

Adoption friction issue can be resolved using the modular layer architecture that Gupta [12] identifies as frequently exceeding technical implementation cost in the enterprise AI deployments. An organization running an existing Kafka event bus can incorporate L2 independently of L1 without changing upstream producers. An organization whose governance framework endorses XGBoost over LSTM mainly for interpretability compliance reasons can deploy L3 in isolation with a quantified accuracy trade-off (AUC-ROC 0.88 vs. 0.91, Table 2) now becomes available to decide on the governance. Replacement of XGBoost with LightGBM or substituting a Transformer architecture for the LSTM requires no modifications to the Flink Application in L4 or the Avro Schema in L1. In live production environments, model improvements are both iterative and continuous, hence, this modularity is not merely an engineering convenience but a prerequisite for a sustainable AI deployment.

Limitations and Threats to Validity

Three limitations require explicit acknowledgement. First, and most significantly, all experiments were conducted on a parametric synthetic dataset. While the distributional calibration approach — log-normal tenure, zero-inflated purchase frequency, 18.3% churn base rate — was designed to reflect published e-commerce benchmarks [21, 22], synthetic data cannot reproduce the schema drift, adversarial manipulation patterns, sudden distributional shifts driven by competitor promotions or macroeconomic shocks, or the long-tail event types that characterise real CRM data streams. Validation on proprietary production data is the indispensable next step before the performance claims of this study can be generalised to specific deployment contexts. This limitation is shared by all prior simulation-based CRM streaming studies in the literature.

Second, integration costs that the adoption-barrier literature consistently found to be substantial are excluded by the experimental protocol – the compliance audits on data governance, authentication, and authorization overhead, model audit trail infrastructure, and the organizational change management investment that Gupta [12] records as often dominating total deployment cost. Pure technical performance under controlled conditions are reflected through the latency and throughput figures reported here which will exhibit higher end-to-end latencies due to these additional layers.

Third, prediction produced without accompanying explanations can be observed through the regulatory frameworks including GDPR Article 22 increasingly require that automated decisions with material consequences be accompanied by specifying meaningful rationale accessible to the affected individual who is subject of the decision [24]. A highest priority technical extension identified from this work is the integration of post-hoc explainability into L4, and SHAP value computation for XGBoost outputs and attention weight extraction for LSTM Sequence scores and its latency cost under streaming conditions requires separate characterization.

Conclusion

In this work, a five-layer streaming artificial intelligence pipeline to proactive Customer Relationship Management was specified, implemented by use of parametric simulation and benchmarked. Using Apache Kafka event capture, Apache Spark windowed feature engineering, an LSTMXGBoost predictive ensemble, an Apache Flink stateful inference dispatcher, the architecture yielded a mean AUC-ROC of 0.91 (SD =-0.006) and F1-score of 0.80 on 30-day churn prediction in ten independent runs - an improvement of 0.10 and 0.11 over a 24-hour batch XGBoost predictor, both statistically significant at p = 0. At 10,000 events/sec, scoring latency was 52 ms with almost linear throughput to 25,000 events/sec (R 2 = 0.998) at processing efficiencies near 99%. The main value of this work is not merely proving the fact that streaming AI is more precise than batch AI, this result is already proven in the component literature. It is proving the combination of the quality of the prediction and the time of delivery is offering the customer a chance to intervene that a batch-cycle architecture just can not offer, structurally, and measuring that with an opportunity to statistically sound conditions like the first time ever. Three research priorities are subsequently followed: careful external validation of proprietary production CRM data in many industries and locations; incorporation of streaming compatible explainability systems to meet new governance and regulatory demands and exploration of privacy-preserving federated learning models that would allow cross-organisation model refinement without the sharing of sensitive customer logs.

References

[1] Sangam GK. Event-driven AI architectures for next-generation CRM platforms. Int J AI BigData Comput Manag Stud. 2026;7(1):49–53. doi:10.63282/3050-9416.ijaibdcms-v7i1p108

[2] Ledro C, Nosella A, Vinelli A, Dalla Pozza I, Souverain T. Artificial intelligence in customer relationship management: A systematic integration framework. J Bus Res. 2025;199:115531. doi:10.1016/j.jbusres.2025.115531

[3] Veeravalli SKD. Integrating IoT and CRM data for unified real-time customer insights via Salesforce Data Cloud. QITP Int J Comput Sci. 2024;4(1):1–16. doi:10.63374/qitp-ijcs_04_01_001

[4] Immadisetti A. Real-time data analytics in customer experience management. Int J Sci Res Comput Sci Eng Inf Technol. 2024;10(6):1280–88. doi:10.32628/cseit2410611172

[5] Ed-Daakouri I, Bouhtati N, Alla L. Big Data analytics in CRM: A systematic literature review. Lect Notes Netw Syst. 2025:423–30. doi:10.1007/978-3-031-88304-0_59

[6] Tung DHLM. AI-powered customer experience: Personalisation and decision-making in CRM. J Econ Stud. 2024;20:55–71. doi:10.52783/jes.1832

[7] Payne A, Frow P. A strategic framework for customer relationship management. J Mark. 2005;69(4):167–76. doi:10.1509/jmkg.2005.69.4.167

[8] Buttle F, Maklan S. Customer Relationship Management: Concepts and Technologies. 4th ed. London: Routledge; 2019.

[9] Leelawathi R, Philip B, Madhusudhanam R, Sony N, Mukthar KPJ. AI-driven CRM: Implementation strategy review. Stud Syst Decis Control. 2024:283–95. doi:10.1007/978-3-031-63402-4_22

[10] Radhakrishnan S, Patel M, Nair V. AI-enabled next-generation CRM: Machine learning approaches for proactive customer engagement. e-Learn Educ Technol. 2025;15(2). doi:10.52783/eel.v15i2.2822

[11] Kumar A, Singh R, Mehta P. Re-architecting CRM systems using predictive analysis and machine learning: An enterprise perspective. Eng Open Access. 2024;2(1):39–50. doi:10.33140/eoa.02.01.04

[12] Gupta AK. Big Data analytics and CRM convergence: Challenges, opportunities, and a structured review. Int J Comput Trends Technol. 2024;72(7):74–82. doi:10.14445/22312803/ijctt-v72i7p109

[13] El Falah Z, Rafalia N, Abouchabaka J. Intelligent data analysis techniques for Big Data-driven e-commerce decision systems. Int J Adv Comput Sci Appl. 2021;12(7). doi:10.14569/ijacsa.2021.0120783

[14] Rainy TA, Rahman MA, Mou AJ. Data-driven decision-making in modern CRM enterprises: A systematic review of latency and model performance. J Econ Theory Value Manag. 2025;38. doi:10.63125/jetvam38

[15] Kreps J, Narkhede N, Rao J. Kafka: A distributed messaging system for log processing. In: Proc NetDB Workshop at VLDB; 2011. p. 1–7.

[16] Zaharia M, Das T, Li H, Hunter T, Shenker S, Stoica I. Discretized streams: Fault-tolerant streaming computation at scale. In: Proc 24th ACM Symp Oper Syst Princ (SOSP); 2013. p. 423–38. doi:10.1145/2517349.2522737

[17] Marz N, Warren J. Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Shelter Island, NY: Manning; 2015.

[18] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proc 22nd ACM SIGKDD Int Conf Knowl Discov Data Min; 2016. p. 785–94. doi:10.1145/2939672.2939785

[19] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735

[20] Chiruvelli KP. Real-time AI for personalised financial product recommendations using ensemble learning. J Inf Syst Eng Manag. 2025;10(60s):68–78. doi:10.52783/jisem.v10i60s.13058

[21] Hooda A. Adaptive real-time Big Data processing: A machine learning scalability approach [preprint]. Research Square. 2024. doi:10.21203/rs.3.rs-4962286/v1

[22] Verbeke W, Dejaeger K, Martens D, Hur J, Baesens B. New insights into churn prediction in the telecommunication sector: A profit-driven data mining approach. Eur J Oper Res. 2012;218(1):211–29. doi:10.1016/j.ejor.2011.09.031

[23] Akter S, Wamba SF. Big data analytics in e-commerce: A systematic review and agenda for future research. Electron Mark. 2016;26(2):173–94. doi:10.1007/s12525-016-0219-0

[24] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM Comput Surv. 2018;51(5):1–42. doi:10.1145/3236009