P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

# Live Stream Analytics for Customer Support and Fulfillment

**Dhanaraj Sathiri**
Independent Researcher
dhanrajsathiri@gmail.com

## Abstract

Data centers serve as the backbone of modern business operations, and their performance is shaped by numerous interdependent factors. Consistent operational quality is essential to meeting the service-level commitments made to customers. Given the finite nature of data center resources, careful planning of support enforcement procedures is critical. These procedures generally fall into two categories, distinguished by the occupancy status of the support service. As a result, the efficiency and effectiveness of both fulfillment and routine maintenance processes are central to delivering reliable service assurance. This research focuses on improving customer engagement management and support service fulfillment.

Meeting service-level agreements (SLAs) and improving overall satisfaction requires strategies tailored to each customer's engagement level. These engagement patterns can be classified based on operational status updates, which in turn influence the choice of fulfillment strategy. Continuous tracking of customer transaction records, paired with real-time alerting mechanisms, is therefore necessary to keep network channels synchronized with backend fulfillment systems. Naturally, the engagement process for a non-occupied customer follows a different pathway than that for an occupied one.

Keywords—Data Center Operations, SLA Fulfillment Management, Customer Engagement Systems, Service Fulfillment Optimization, Maintenance Process Management, Resource Allocation Strategies, Occupancy-Based Service Models, Customer Interaction Tracking, Real-Time Service Monitoring, Alerting Mechanisms, Backend Service Synchronization, Operational Efficiency, Service Quality Assurance, Customer Satisfaction Optimization, Workflow Orchestration, Service Lifecycle Management, Transaction Monitoring Systems, Support Process Automation, Data Center Service Analytics, Engagement-Driven Operations.

## I. Introduction

Most data center operators maintain an online service availability higher than 99.9% to satisfy demanding service level agreements (SLAs), but disruption can still threaten customer engagement and confidence. A common practice is to set up a contact channel (ticketing system or chatbox) for user-initiated support, which is usually asynchronous and provides an implicit indication of an incident. However, customers often refrain from making contact when an issue arises. The services provided by the data center may also depend on third-party services that are outside the operator's control. Instead of pushing alerts when conditions require them, the tickets can be treated as inquiries. During a customer support interaction, live details regarding ongoing order fulfillment or service provision processes can be retrieved and served to the agent or user in real time to speed up service resolution.

Adapt flexibility in ticket handling for better customer support and follow two fulfilling orders process-tracking principles. An agent might not have the answer to the user's question when the ticket is assigned—yet delivery of the requested information may be time-sensitive. Therefore, full personalization of the offered support for such users is required.

## II. Theoretical Foundations of Real-Time Stream Analytics

Implementation of real-time stream analytics in data center operations requires consideration of real-time

P-ISSN: 3050-9734 E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

processing concepts. Real-time refers to processing operations on a data stream in such a way that the same response time constraint is guaranteed and enforced on the processing. The concept of time plays a big role in this type of environment since data move continuously into the processing engine over time, either ordered by the sending order or based on the event-generation time. Therefore, different processing and logic techniques are used per case. Like any message, data records have assigned a header that contains information needed by the various systems to which it connects. The primary stream messages are events; these messages do not just contain data values but respond to state change in the monitored environmental premises. Streaming data are messages generated permanently by one or multiple publishers while streaming logical refers exclusively to the processing of these messages by the border nodes.

Streaming data can be processed in event time by the special and more complex query techniques based on event-time windowing. Windowing allows the division of the complete set of data generated over a period into manageable subsets that can be processed sequentially. In these cases, it is necessary to consider that the limits of these subsets are always based on the event generation time. It is clear that data arriving generally out of order due to various reasons, such as network delay, should be tolerated over a certain acceptable time distance. Latency, reliability, and throughput are the main parameters that have to be considered when analyzing a service-quality assurance problem. Latency is the time delay between the generation of an event and the delivery of the corresponding response. Latency has to be controlled to respect a given service-level objective, whereas throughput is the average number of messages sent by a one-to-many consumer in a unit of time.

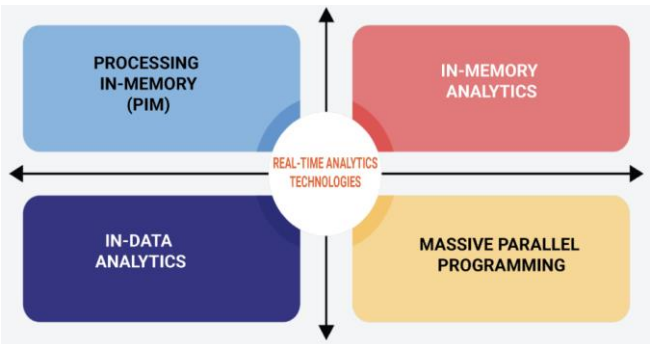



**Fig 1: Real-Time Analytics Technologies**

*A. Essential Concepts of Real-Time Stream Analytics*

Real-time stream analytics enable the detection, analysis, and exploitation of potentially critical events as they occur or soon thereafter, with minimal delay. The event time of the data indicates the moment in history to which the data element belongs, independent of when it gets collected and processed. In an event-time processing model, the event-time granularity of data remains constant, creating the potential to provide accurate quality-of-service guarantees on latency and throughput. Windowing techniques allow computations to be conducted over finite partitions of the input stream to sustain acceptable performance. A window can be delimited by data values or time duration, and multiple windows can be active on the same stream at the same time, with different windows producing results over the same input data at different latencies, thus favoring different quality-of-service levels.

Performance metrics, such as latency and throughput, provide a qualitative measure of the ability to handle potentially irregular, volatile, and unbounded input data volumes. Latency is defined in relation to the event-time source, meaning that observed quantities can change only with the arrival of new data, while throughput defines the number of separate events produced by the system. The analysis quality depends on the data volume—greater amounts of data improve detection quality—while a system sensitive to a greater volume of changes produces lower quality of detection. In a data center, quality-of-service hours to customers from different operational areas must match to avoid SLA violation risks.

***Equation 1. End-to-End Latency***

Let:

$$T_g = \text{event generation time } T_d = \text{event detection/response time}$$

Latency is the delay between generation and response:

$$L = T_d - T_g$$

For many events:

$$L_1 = T_{d1} - T_{g1}\ L_2 = T_{d2} - T_{g2}\ L_n = T_{dn} - T_{gn}$$

Average latency:

$$L_{avg} = \frac{L_1 + L_2 + \cdots + L_n}{n}$$

P-ISSN: 3050-9734 E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

Compact form:

$$L_{avg} = \frac{1}{n}\sum_{i=1}^{n}(T_{di} - T_{gi})$$

Article result:

$$L_{avg} = 24.86 \text{ seconds}$$

## III. METHODOLOGY

A conceptual design serves as the methodological framework for the study. Data exchanges with external service providers and partners deploy the Representational State Transfer protocol through the HTTP (HyperText Transfer Protocol) networking protocol. Support tickets that customers create are provided by a web interface that is integrated with a database. The database maintains customer information and also houses recorded support ticket information. The database is capable of storing services and stock items provided by the data center and is integrated with a web interface for easy access.

Real-time analytics support direct communication with customers by delivering end-to-end insights into fulfillment activities. Order fulfillment touches for every customer order have predefined time constraints that depend on contract negotiations, but order fulfillment services need not follow a strictly routed path. A combination of customer demands and inventory availability in the warehouse render some routing choices more preferable than others, and such routing choices are capable of being specified in a dynamically optimized manner as a joint optimization problem. Consideration of such optimization choices leads to exploratory definition of a Latency-Aware Routing and Fulfillment Optimization (LARFO) service. Other types of considerations are driven by inventory stock management. While a stockitem must be sourced to a customer from an appropriate facility, careful demand prediction and imminent stock-out risk assessment decisions can be made.

**TABLE 1: SYSTEM PERFORMANCE METRICS**

| Metric Name | Symbol | Value | Unit | Description |
|---|---|---|---|---|
| Daily Requests | $N_d$ | 300,000 | requests/day | Total incoming customer requests |
| Event Records | $N_e$ | 4,000,000,000 | records | Stream data volume processed |
| Average Latency | $L_{avg}$ | 24.86 | seconds | Time delay from event to response |
| Uptime | $U$ | >99.99 | % | System availability |
| Servers | $S$ | 1000+ | count | Infrastructure scale |
| Processing Nodes | $N_p$ | 4 | nodes | Physical processing nodes |

### *A. Dynamics of Customer Engagement and Order Fulfillment Strategies*

Data center customers engage with service providers along different channels and in various stages of their customer lifecycle. For help-desk support through dedicated tickets, a service level objective prescribes the maximum response time, up to which the technology provider is obliged to acknowledge and comment on the ticket. Apart from explicitly raised incidents, customers often signal other inquiries through unstructured communication. These channel- and time-dependent signals also define the stage of the incident lifecycle. Addressing them through proactive solutions at an appropriate level of detail avoids additional friction. Properly interpreted support inquiries further improve customer experience and help build a loyal brand community.

Both live order processing and inventory synchronization touch on ticket escalation. Notification of planned or perceived downtime involves specific customer wishes. On the demand side, a service-level objective prescribes the maximum time allowed for the delivery of any computation. For support handling, proximity to real-time data center status remains an important competitive factor. With multiple locations, fulfillment routing also represents an important cost factor. To both minimize shipping time and reduce cost,

P-ISSN: 3050-9734 E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

routing decisions depend on the distribution of demand and timely estimation of delivery latency. Intelligent fulfillment engines dynamically allocate orders to the logistics network at different levels, directing them through either the fastest or cheapest path.

***Equation 2. Throughput***

Let:

$$N = \text{number of processed events } \Delta t = \text{processing time interval}$$

Throughput means events processed per unit time:

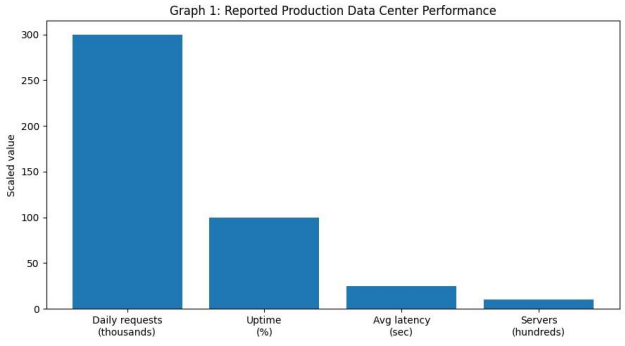
$$TP = \frac{N}{\Delta t}$$

If the system processes 4 billion records:

$$N = 4{,}000{,}000{,}000$$

Then:

$$TP = \frac{4{,}000{,}000{,}000}{\Delta t}$$




## IV. Objective of Study

The amplification and coordination of fulfillment workflows through ordering and customer support channels represent the two main interacting functions in a data center operation. The objective is to define and realize a data-driven adaptive model of interaction that will enable these two areas to augment their respective processes for cost savings and assurance of service.

Four major hypotheses can drive the research: (1) The adaptation of customer support processes to real-time contextual signals enables more efficient and targeted assistance. (2) The continuous operation of customer request life cycles provides additional information for anomaly detection, reduces MTTI, and ensures maximized MTRS. (3) Orders recorded on an operations management platform provide additional contextual signals to be preserved, predicted, and actively synchronized. (4) The interaction scheme among the customer support and order-monitoring functions creates an incremental practical model of adaptive support and inventory synchronization, fulfilling the requests of direct end users and service management in a coordinated way.

### *A. Data Collection and Event Processing Mechanisms*

Data collection and event processing are crucial aspects in the context of real-time stream analytics for adaptive support and order fulfillment in data centers. The various data sources support numerous third-party applications, which follow the predefined data source interfaces and communicate over the HTTP REST protocol. The JSON data format is used to transport the event data and transmit every relevant context data, optimized for bandwidth, that are streamed into the system. The following paragraphs describe the event schemas and data collection mechanism of the data sources. The actual data transport, streaming pipelines, analytics engines that ingest or publish on those data source interfaces are not covered in this section. The actual data transport operations are highlighted in the sections ahead, matched with proper streaming platforms and processing paradigms that are responsible for JSON data transport, transport layering, buffering, and dealing with data schema evolution which happens frequently in practice.

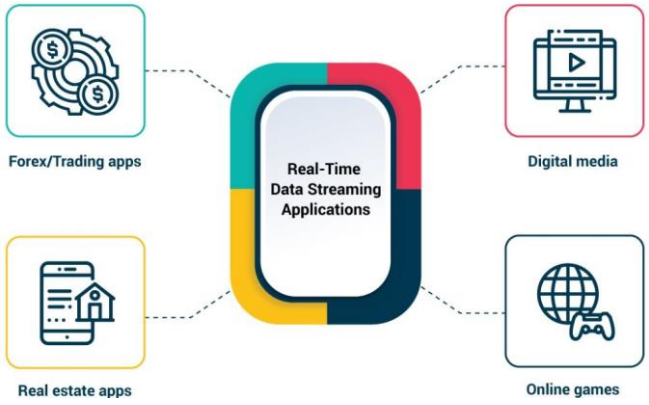
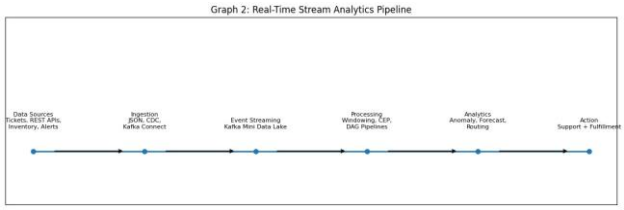



**Fig 2: Real-Time Data Streaming Applications**

P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

A data collection system of the data source interfaces streams the observed events into a real-time mini-data lake build on Apache Kafka. Additional data sources are periodically ingested or bulk loaded into the mini-data lake over a longer interval on an hourly basis, daily basis, a weekly basis or as required. The all-data mini-data lake serves as a back-end for previous event-time processing enabling dependency on event time progression and speeding up the overall process through more resource-efficient yet sufficient support nodes. The additional data sources enhance the customer support analytics, asset service dependency tracking, digital-marketing persona and preference update for real-time personalization, and other analytics that are sensitive to event-time progression.




## V. Research Summary

The theoretical foundations, methodology, and technology stack for a streaming analytics architecture applied in the context of data center operations are introduced. Real-time signals from public clouds are harvested and streamed through a cloud-native architecture. The operational requirements of real-time interactions between customers and service providers are also presented, along with the objectives of data processing. The real-time analytical system provides contextual cues that improve customer profiling, helping to understand current interactions, anticipated needs, and ongoing issues. Recent incidents are also considered, detecting anomalies in monitored signals, conducting root-cause analysis using alerts from third-party components, and accelerating remediation. Supporting decision-making capabilities are included as signals evolve outside expected ranges.

A stream processing engine consumes weather and traffic conditions for responsive fulfillment routing, an inventory management system detects replenishment needs to align stock levels with sales forecasts, and a stock-out risk estimator calculates the probability of empty shelves. Core functions identify the optimal delivery strategy, minimizing latency by identifying the closest provider with sufficient stock, distributing multiple orders to the same address, and routing multi-stop journeys to maximize vehicle utilization. Cooperation with fulfillment services ensures appropriate timing and service level guarantees. Despite expected delays, customers are informed of real-time provisioning using preferred communication channels. These functions address availability, adaptive support, and completion service level objectives.

### *A. Real-Time Customer Engagement and Contextual Insights*

Signals from real-time customer interactions with data center services—such as service status updates, ongoing tickets and requests, and detected and reported quality anomalies—feed mechanisms of real-time stream-based analytics to deliver personalized, contextually relevant support interactions adapted to the customer situation, inferred preferences, and available resources. On the support staff side, triggers are detected and served as potential incidents and emergencies requiring immediate attention. Support staff are thus able to focus on fulfilling the needs of customers who need assistance most urgently. Consequently, customer engagement is optimized to maximize the quality and speed of resolution of submitted service requests.

Customer interactions are analyzed to determine personal and group attributes useful for future interactions and to predict what kind of help customers might need in future engagements. Automated monitoring systems sensing data from a multitude of sources detect possible anomalies disturbing the quality of provided services. When customers initiate new service requests, their ongoing situation and the past status of the service, correlating detected anomalies, and customer affinities are used to customize the interaction and provide better, more personalized help.

## VI. Data Center Context: Customer Support and Order Fulfillment Requirements

Operational requirements for data center service providers encompass imperatives for high uptime when fulfilling customer requests and for correctly coordinating the cross-functional activities of customer support and order fulfillment departments. The two departments are primarily engaged in different activities at different points of time. Customer support focuses on resolving issues related to the products and services that customers actually use; order fulfillment primarily deals with incoming supply requests, such as the procurement of new hosts, network connections, and other cloud resources, along with the subsequent provisioning of those requests and the configuration of associated services. However, these departments do cross

P-ISSN: 3050-9734 E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

paths during incident handling when support issues cause customers to raise complaints or requests for refunds on their service credits. In addition, these departments need to communicate with each other during the transition from an ongoing service-level agreement (SLA) to a service-level agreement support (SLAS) arrangement, in which an external customer uses the data center infrastructure and internal data center operations personnel provide support. There are also other touchpoints during the customer engagement journey when the support department needs to proactively reach out to customers to predict SLA breaches and offer corrective actions, or when order fulfillment needs to proactively seek assistance from customers to complete a task in the hosting and provisioning workflows.

Across these engagements, the customer support department—which facilitates fast issue resolution to increase customer satisfaction and retention—and the order fulfillment department—which fills requests seamlessly to increase order fulfillment speed and customer satisfaction—have service-level objectives (SLOs) that are reflected in the guarantees offered to customers. Data centers operating on a retail basis restrict their offered guarantees to start dates of planned connectivity requests, a minimum booking horizon of planned connectivity requests, and a preestablished maximum of planned active connections. These engagement patterns and operational hibernation support constraints need to be reflected in the strategy for providing data to data analysts and data scientists during the decision-making phase of the analytics value chain.

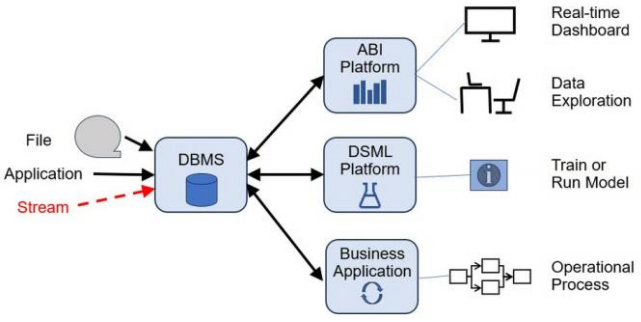



**Fig 3: Stream-Enabled DBMS**

*A. Customer Interaction Dynamics and Fulfillment Strategies*

Customer interactions and service fulfillment touchpoints are envisioned to facilitate an effective orchestration of data center operations. Such engagement channels and related service delivery mechanisms are closely aligned with customer service supply-chain management principles, focusing on the definition of service guarantees based on customer need states and the fulfillment of service commitments in a timely manner. Consequently, services are associated with two key aspects: the dynamics of engagement throughout the customer use of the service and the specific fulfillment strategies for each service area.

The dynamics of customer interactions with the service provider can be understood in relation to customer lifecycle management principles, which promote customer engagement as a strategic lever for creating and maintaining lasting relationships with customers. These principles recognize the existence of different stages throughout the customer lifecycle, with specific strategies for each stage. To a certain extent, the same quality of service dictates the interaction strategy with the customer, considering that every service has a time element at its core. One of the main service level objectives (SLO) of the data center operator is to offer high availability of the service infrastructure, fulfilling the commitments with suppliers and customers. In this context, customer interaction can be classified into three different stages: proactive, active, and reactive, with different operational objectives along these lines, which need to be considered in conjunction with the fulfillment of service delivery activities.

**TABLE 2: REAL-TIME STREAM ANALYTICS COMPONENTS**

| Component Name | Layer Type | Function | Technology Used |
|---|---|---|---|
| Data Sources | Input Layer | Generates events | REST APIs, Tickets, Sensors |
| Ingestion Layer | Collection | Captures incoming data | JSON, HTTP |
| Event Streaming | Transport | Streams real-time data | Apache Kafka |
| Processing Engine | Computation | Processes streams | DAG Pipelines, CEP |
| Analytics Layer | Intelligence | Detects patterns | ML, Anomaly Detection |

P-ISSN: 3050-9734 E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

| Component Name | Layer Type | Function | Technology Used |
|---|---|---|---|
| Action Layer | Output | Executes decisions | Support + Fulfillment |

## VII. Architecture for Real-Time Analytics in Data Centers

The architecture comprises four primary layers that facilitate the comprehensive capture of signals across customer journeys. These inputs are combined through event-stream processing to provide insights and auto-responses, establishing two-way communication with customers and order-fulfillment teams. The customer-engagement surface, hosting digital assistance, interaction histories, and admin interfaces, serves as the final destination for the combined insights. Within the operational environment, the Admin and Console module supports order fulfilment and travel planning for on-site personnel.

Signals originating from customers, fulfilment, or inventory require suitable event-streaming technology for instantaneous delivery. A feedback channel streams feedback events to the Capture module, where the Data-Collection layer provides suitable interfaces for any source system. The Event-Streaming layer employs a data-in-transit modelling approach, incorporating required buffers to deal with bursts in video and call-logging traffic and to ensure data-schema evolution remains resolvable throughout the overall stream processing design. All other real-time data-collection channels support event schemas that use established plug-and-play protocols for direct integration.

Real-time analytics now continuously analyses these data streams to generate context-specific, forward-looking insights for customer-engagement-personalization teams and order-fulfilment personnel. Anomaly detection or root-cause-analysis systems form the first layer of the combined insights, facilitating appropriate remediation before issues escalate. Services or products are automatically replaced or offered at significant discounts in the event of detected components nearing unavailability. The second layer generates user models that continually track demographic parameters, user actions, and interests detected in content accessed through primary digital-engagement channels, enabling automatic adaptation of all visible services.

***Equation 3. Uptime Percentage***

Let:

$T_{up}$ = available time $T_{total}$ = total observed time

Availability:

$$A = \frac{T_{up}}{T_{total}}$$

Percentage uptime:

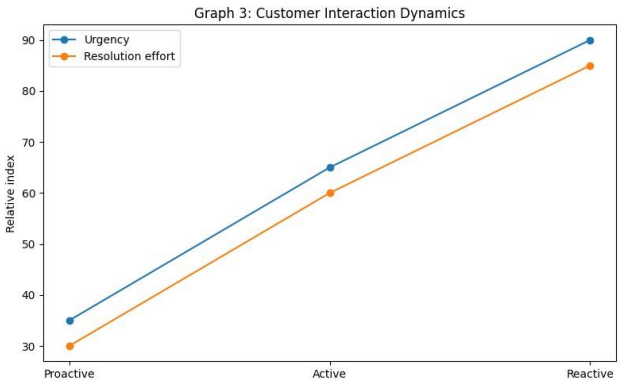
$$Uptime = \frac{T_{up}}{T_{total}} \times 100$$

Article states uptime is over:

$$Uptime > 99.99\%$$

Downtime percentage:

$$Downtime = 100\% - 99.99\% \; Downtime = 0.01\%$$




### *A. Data Ingestion and Event Streaming*

Data flows into the architecture through multiple sources. Data connectors continuously or on-demand ingest events from cloud and collocation data services via the simple object access protocol (SOAP) and Representational State Transfer (REST) Application Programming interfaces (APIs) into a staging area. An unattended process converts textual data from a knowledge management system API using the JavaScript Object Notation (JSON) format into a durable storage buffer in JSON format. Data from collocation service personnel and a support team engaged in an incident are sent to the enterprise support exchange's support automation system, captured in a JSON format via a Web application and redirected to the durable storage buffer. The operation of network element isolation detection systems generates alerts that are captured in Change Data Capture (CDC) format using

P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

Kafka and stored in the system's atomic log. Scoring engines in the platform process cloud and collocation scratch accounts at defined intervals and send the logs to the durable storage buffer.

Sources serving event streams to the architecture conform to the Kafka Connect framework. Events pass from the durable storage buffer into OpenSearch's buffering layer, where the stream processing engine and rules engine are hosted as independent microservices. Events are processed using a stream-processing paradigm based on the Kappa architecture pattern. The stream-processing engine receives events from the buffering layer and routes them to dedicated computational pipelines, providing scalability and performance benefits to the architecture's design. The computational pipelines exploit the stream-processing engine's capacity to scaffold logical execution trees through the orchestration engine. The directed acyclic Graph (DAG) offers fault-tolerant, parallel, and stateful capabilities with minimal effort. Stream-processing computations employing non-event-time information are executed in micro-batches. Integration with cloud-provisioned assets is achieved through the serverless patterns of cloud providers, enabling the execution of infrequent yet systematic bulk-data operations.

### *B. Stream Processing and Computational Pipelines*

Prioritization of event order, delivery guarantee, and parallel processing capability shapes the choice of stream processing engine. Processing of event streams in a single flow or logical sequence avoids undue complication of the solution and eliminates need for complex data-flow specification. Stream processor capabilities inform the design of individual computational paths. Individual paths handle non-overlapping processing tasks and thus also allow a deployment strategy that assigns each path to a separate worker instance. Computation may thus involve any combination of partitioning and shared access, whether parallel-processing requirements exceed resources of single deployment node or sit comfortably within those resources.

While a data lakehouse is not strictly required for real-time processing and analytics, the combined asset-management and streaming architecture fits the needs of multiple data- and workflow types. Cross-connecting sensor and event streams in near real time, satisfying end-to-end latency and throughput requirements, and duly processing and storing supporting data and assets all contribute to ensuring round-the-clock service availability and aligning delivery commitments with operational capability. What-If analysis may predict the effect of critical-path workloads on Service-Level Agreements.

***Equation 4. Event-Time Window Average***

Let event values be:

$$x_1, x_2, x_3, \dots, x_n$$

For a window of size $w$:

$$W_t = \{x_{t-w+1}, x_{t-w+2}, \dots, x_t\}$$

Window average:

$$\bar{x}_t = \frac{x_{t-w+1} + x_{t-w+2} + \cdots + x_t}{w}$$

Compact form:

$$\bar{x}_t = \frac{1}{w} \sum_{i=t-w+1}^{t} x_i$$

## VIII. Adaptive Customer Support Through Real-Time Insights

The diversity of real-time context signals and the corresponding capabilities required for customer support signal, modify, and enhance the incident support lifecycle. Whenever a user indicates service-related issues—for instance, through messages sent to a cloud provider's support service chat—real-time signals about status or incidents can help provide personalized service while assisting support agents and avoiding repeated interactions. Context signals assist in two ways: they help create a model of each user that incorporates aspects such as actual/future usage, preferences, and the overall incident landscape. Subsequently, these evolving models can provide indications that inform the agent or automated service directly about the root cause, status, and likely resolution of the user incident.

First, models capture usage dynamics for each user. Signals such as cloud product monitoring views, usage forecasting for dedicated hosts, and chat or call usage provide bases for creating statistical usage models. Whenever servers, networks, databases, or storage elements display anomalies that may impact service, the techniques not only warn about foreseen problems but also indicate users possibly affected. At any moment, inferences about preferences can be sought from social networks and logged service interactions. These inferences contribute to dynamically endowed automatic

P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

responders and also aid support agents. Potentially starting with frequent user questions and expanding with questions regularly replied to by support, the automatic service is likewise offer able to respond in natural, informal language.

Second, signals can assist direct support personnel with the analysis of user issues. Whenever users make inquiries or signal problems, techniques analysing cloud status information can indicate whether the problem is an ongoing or imminent incident and provide supporting details. Techniques assimilating past support and public-user responses can reveal potential root causes and timely solutions; notification analyses can bring up connections with similar incidents. If the problem remains unresolved for a defined time period, prevailing usage models of the user can trigger notifications incorporating indications of the problem, such as suggested reasons for delays in support response.

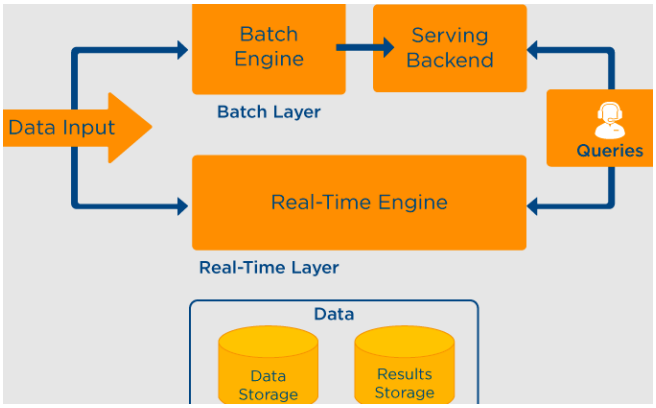
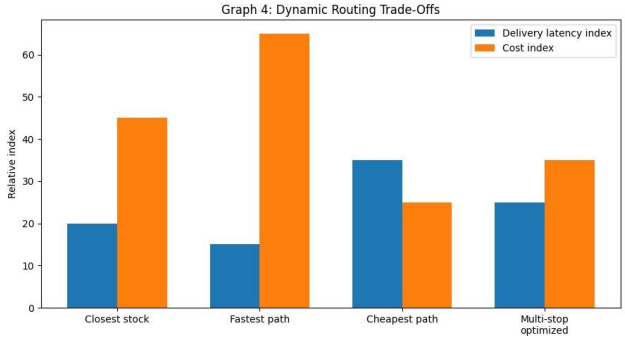



**Fig 4: Streaming Analytics Architecture**

*A. Real-Time Personalization and Context Awareness*

To avoid overwhelming customers and causing frustration through irrelevant or excessive offers, it is essential to personalize assistance while maintaining readiness to offer meaningful support. A user-centric model can represent each individual, taking into account their signals, past interactions, and potentially useful contextual cues for future assistance. Certain fine-grained actionable signals can also inject a sense of urgency into relevant action links that aid in quicker resolution of on-going incidents. Sources of personalization cues that can be used for aid or new product recommendations include access and usage reports, current live issues and their assessment, and stock or service availability information.

Current live signals can also trigger assistance offer prompts or conclusion invitations for on-going interactions. Customers with open incidents requiring diagnostics or corrections may appreciate such prompts. Concluding an open interaction can also allow the user to easily finalize an impending status update for service impact communication. Since change requests have various sets of activities involved, fulfilling these requests may require resources from various teams. However, during peak request times, an activity may be delayed due to resource unavailability. A change request requires a follow-up closure on user side to ensure the action/downtime has taken place according to a request. Thus, assistance prompts linked to these rationales can also be activated in context.




*B. Incident Detection, Diagnosis, and Resolution Acceleration*

Incident detection, diagnosis, and resolution are both necessary and time-sensitive responses to system deviations that reliable infrastructures must address at the system, service, and support levels. Inadequate, ineffective, or delayed detection and response result in dissatisfied customers, and unsatisfied customers are ordinarily the first to defect when better alternatives emerge in the market or become affordable.

Real-time analytics can therefore inform support-guiding interactions by streaming user contexts, unanticipated or persistent issues, and pre-existing solutions. These three sources of information can be monitored independently. Anomalies are detected through classical statistical process control techniques. For persistent anomalies, recommended users resolutions are drawn from knowledge bases constructed for previous occurrences. Whatever the context of the user, a detailed diagnosis and solution are suggested to assist the customer support engineer dealing with the request.

P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

The faster the resolution can be executed, the greater the time saving for the CSE.

## IX. Real-Time Order Fulfillment and Inventory Synchronization

Focus on the end-to-end workflow linking demand signals with supply responses: how live indications of risk exposure activate preventive actions through the fulfillment chain. A suitable order fulfillment operation must satisfy customer needs while optimizing the resources involved. Fulfilling customer orders typically means expediting deliveries from stock, manufacturing, assembling, or sourcing products. However, stock-outs result in longer delivery times or the non-delivery of products, which displeases customers and diminishes loyalty. Preventing such costly failures also incurs costs, which are related to the safety stock levels that ensure an adequate service level while avoiding excessive expense proportions. Consequently, inventory must meet demand while incurring a minimum expected total cost. The ideal to satisfy a customer and minimize costs is to ship the product from the closest stock location. Therefore, routing orders and shipments should prioritize physical distances and speed. A customer usually expects to receive a fulfilled order no later than the specified service level.

Live inventory visibility and demand forecasting produce useful predictive signals. Individual product stock positions at each location are updated for each replenishment and consumption, while the sales history predicts future demand. Traffic conditions predict the actual delivery times between any two locations in the network to facilitate the decision of which location must fulfill every customer order. When stock-out risk evaluations trigger such actions, these predictive signals process fulfillment operations over the supply chain. Candidates for routing candidates join the resources tracked for routing decisions. Mitigation of stock-out risk ensures that the customer receives the product in the guaranteed period, helping also maintain customer loyalty.

### *A. Live Inventory Tracking and Demand Forecasting*

Live monitoring of IT hardware inventory and smart yet efficient anticipation of future needs is important for ensuring that adequate stock levels are maintained, especially for spares that have long lead times. Real-time visibility into spare inventories for different servers, network switches, routers, and so on enables operations teams to fulfil customer orders immediately. Demand from customers for such spare parts, if predicted several weeks in advance, helps for predictive replenishment of the same. Missing any such predicted demand leads to unnecessary delays in delivery, since, from an order placement to order fulfilment, some parts require a month or longer just for delivery, even if other actions can be executed immediately.

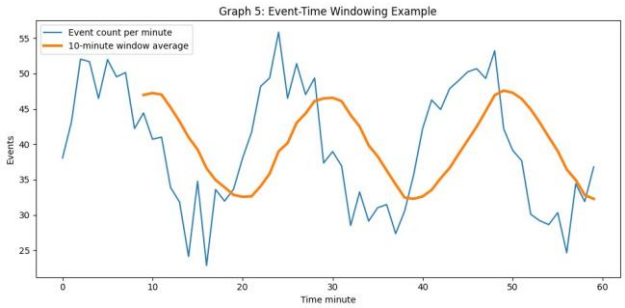
Hardware parts that are either speculated to go out-of-stock in the near future, or those parts that have already gone out-of-stock, yet also remain in demand, should preferably be avoided. For such parts, awaiting receipt of inbound parts should preferably be accommodated. Such smart monitoring and anticipation reduces stock-out situations significantly; however, occasionally missing certain rush demand might still be unavoidable. A rush order can always have a slightly higher fulfilment latency compared to other orders if the routing is performed right. If the rush order can be served when moving the shipment to the main delivery centre, then all the better; else, a swap in allocation will ensure low latency.

### *B. Dynamic Routing and Fulfillment Optimization*

Based on the detected demand and current fulfillment status, routing decisions taken in a customer interaction may have corresponding internal actions or communications after a pre-defined investigation and decision-making, internal response may include an e-mail on out-of-stock stress template for production or buying fulfilment teams, or allocation may propose a dedicated flow connection might be or not a connection to shorten delivery time. If at least one larger order in forward connection not yet processed is detected, the fulfillment team must deliver quantity hardware system-bound at lowest latency to a larger order, independent of stock state, best when stock shortage is detected.

Demand signals also modulate hardware system-centred control, forwarding a predicted order quantity in future time or direct to stock availability to determine use-case, such as for telecommunications or back-up plan routing. With stock availability stream signals the system also tracks connections and creates opportunity fulfillment location node. Connection creating opportunity routes outside the service area node or long latency is always negated.

P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18




***Equation 5. SLA Violation Condition***

Let:

$L$ = actual latency $L_{SLO}$ = maximum allowed latency

No violation:

$$L \leq L_{SLO}$$

Violation:

$$L > L_{SLO}$$

Violation indicator:

$$V = \begin{cases} 1, & L > L_{SLO} \\ 0, & L \leq L_{SLO} \end{cases}$$

## X. Data Quality, Governance, and Compliance in Streaming Environments

High data quality underpins analytics confidence, operational performance, and business value. Systems consider the influence of data on analytical quality as well as the end quality of analytics-derived data stores. Quality management defines domain-specific data quality metrics, monitors quality via automated testing and profiling, reconciles issues, codifies controls, and coordinates processes, roles, responsibilities, and procedures for ensuring ongoing quality. Classical data governance expands data quality to cover provenance, stewardship, access permissions, operationalization processes, and compliance with privacy, security, and regulatory frameworks. Governance incorporates and extends data quality through data lineage management and support for operationalizing analytical models. Although often identified with batch-style reasoning, data quality and governance are equally valuable in streaming environments, as illustrated by the fulfillment pipeline.

Quality management concentrates on ensuring the accuracy of dynamic activity tracking and availability of key infrastructure components. Data governance focuses on regulating architecture maintenance to reflect current operational usage and access control, monitoring data quality during processing, and capturing data lineage to support regulatory compliance.

Data quality is administered using a combination of validation rules and automated data profiling across production sensor streams. Validation tests enforce expected ranges and frequency distributions. Data profiling detects emerging data quality issues, such as missing data or schema drift as a result of changes to the monitored infrastructure. Automated profiling operates over a configurable time window and alarms when detected anomalies exceed defined thresholds. Provenance tracking records how created, derived, or consumed data relates through standard lineage graphs. Such lineage ensures appropriate auditing and modeling of personal identifiable information flow and retention, enabling compliance with relevant privacy and data protection regulations such as the EU General Data Protection Regulation (GDPR).

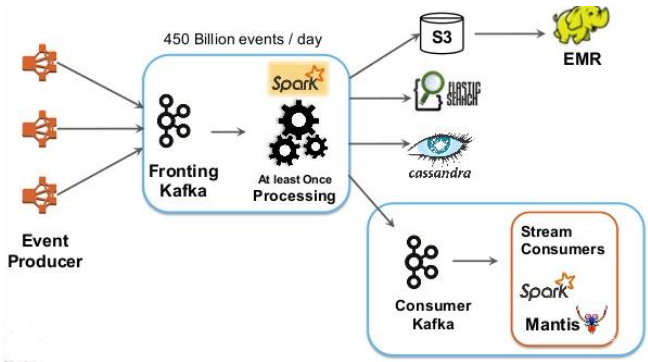



**Fig 5: Data Streaming and Stream Proce**

### *A. Ensuring Data Integrity and Regulatory Compliance in Stream Processing*

Construction of Proper Data Quality, Provenance, Governance, and Regulatory Aspects for Data in Streaming Environments

Stream-based systems often blend data from multiple sources, necessitating strict cross-checking and validation mechanisms to guarantee uninterrupted quality. Ingestion pipelines must assure that new tuples conform to required quality standards, preventing defects from propagating unnoticed through subsequent processing and analysis tasks.

P-ISSN: 3050-9734
E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

The source and origin of each tuple ideally should be recorded in an accessible fashion, along with enforcing mechanisms for data quality controls, governance, and compliance with government directives in specific application domains, such as user privacy and data protection or company-level data management and governance policies. Data validation rules must be defined according to the schema of the event that is being used by application, either internal ACID rules or business rule defined by business policy. Rules for data lineage must also be defined in context of business policy considering the organization needs and environments. Privacy restrictions should also be applicable to master ingress data or control parameters, according to certain conditions.

Preventive modules can be moved to the front of the engine in the future during the design phase of certain applications. A preventive module handles preventive action for control parameters or state changes at different layers. For example, entering a prediction mode would suspend alarm detection or a critical condition check, even if such changes could have been dynamically predicted. Dynamic visualization can suppress cache misses for data rendering, either continuously or upon explicit request.

**TABLE 3: CUSTOMER INTERACTION LIFECYCLE MODEL**

| Stage Name | Symbol | Characteristics | Objective |
|---|---|---|---|
| Proactive | $C_p$ | Early engagement | Prevent issues |
| Active | $C_a$ | Ongoing interaction | Resolve requests |
| Reactive | $C_r$ | Post-issue response | Fix incidents |

## XI. RESULTS

The analytic framework was deployed in a production data center environment that receives 300,000 requests daily from diverse customer segments. Uptimes of over 99.99% and complex support workflows, triggered by events and across multiple channels, create conditions for adaptive personalization based on customers' operational context. The architecture was designed with these high availability and cross-functional requirements in mind, and performance was evaluated from three perspectives. First, the latency, throughput, and detection accuracy of the system were measured. Second, customer support interactions and system components for which real-time context could increase service efficiency or reduce handling effort were identified. Last, the impact of real-time fulfillment signals on order latency was investigated.

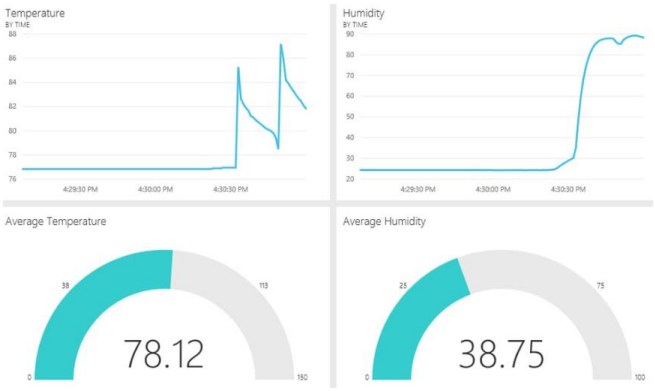



**Fig 6: Streaming Analytics**

For the first perspective, three processing pipelines for traffic ingestion and analysis served as prototypes. Using an event stream comprising 4 billion records from over 1,000 servers, an open-source stream processing engine ran on two virtual machines across four physical nodes in a Mesos cluster. Metrics included end-to-end latency (e.g., from traffic generation to event detection), throughput, fault tolerance, and prominence in incident mitigation strategies. Results indicated an average end-to-end latency of 24.86 seconds; the main bottleneck was machine performance for network traffic analysis, which a future integration with Apache Spark could improve.

## XII. CONCLUSION

Data center operations deliver services with the availability and support requirements of mission-critical systems. In this environment, real-time stream analytics enables companies to realize adaptive customer support and synchronized inventory control. By detecting events across different products and services and matching them with customer interaction models and service guarantees, these methods determine the context and content of (internal and external) support interactions as well as what order fulfillment and distribution routing actions to trigger. Such services improve user satisfaction while mitigating downtime costs for data center operators.

Five design principles guide the application: (1) Quality of Service requirements for an internal customer audience, with Natural Language Processing (NLP) powers for both

P-ISSN: 3050-9734 E-ISSN: 3050-9742

EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES

VOLUME: 04 ISSUE: 01

RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

textual and voice conversation channels; (2) a combination of pipelining and complex event processing (CEP) to cater to the expected end-to-end latency; (3) the integration of monitoring and control systems for rapid detection, diagnosis, and resolution of incidents; (4) support for a variety of connected devices that frequent updates of device and network context can leverage to enhance the user experience; and (5) the processing of data from inventory systems deploying machine-learning algorithms to avoid traditional stock-out scenarios.

## REFERENCES

1. Wang, J., Wang, J., Sun, C., Li, S., Liu, X., Si, L., Zhang, M., & Zhou, G. (2020). Sentiment classification in customer service dialogue with topic-aware multi-task learning. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 9177–9184.
2. Jia, Y. (2020). A deep learning system for sentiment analysis of service calls. Proceedings of the 3rd Workshop on e-Commerce and NLP, 24–34.
3. Paleti, S., Baliyan, M., Aitha, A. R., Reddy, B. A., Bhadauria, G. S., & Sing, S. A. (2025, August). Graph—LSTM Hybrid Model for Improving Fraud Detection Accuracy in E-Commerce Financial Services. In 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS) (pp. 1-6). IEEE.
4. Borg, A., & Boldt, M. (2020). Using VADER sentiment and SVM for predicting customer response sentiment. Expert Systems with Applications, 162, 113746.
5. Kolla, S. H., & Mangala, N. (2025). DESIGNING AUTONOMOUS LLM AGENT FRAMEWORKS USING GEN AI PIPELINES TO ENHANCE CUSTOMER SERVICE MANAGEMENT AND KNOWLEDGE WORKFLOWS. Lex Localis-Journal of Local Self-Government, 23, 9719-9733.
6. Feizabadi, J. (2020). Machine learning demand forecasting and supply chain performance. International Journal of Logistics Research and Applications, 24(2), 119–142.
7. Følstad, A., & Taylor, C. (2021). Investigating the user experience of customer service chatbot interaction: A framework for qualitative analysis of chatbot dialogues. Quality and User Experience, 6, Article 6.
8. Ngai, E. W. T., Lee, M. C. M., Luo, M., Chan, P. S. L., & Liang, T. (2021). An intelligent knowledge-based chatbot for customer service. Electronic Commerce Research and Applications, 50, 101098.
9. Mangalampalli, B. M., Bandi, V. D. V. K., Kolla, S. K., & Kumar, M. V. K. (2025). Towards Self-Evolving Healthcare Intelligence: Integrating Advanced Learning Systems with Real-Time Clinical Data Pipelines. Cultura: International Journal of Philosophy of Culture and Axiology, 22(12s), 464-486.
10. Cattermole, A., Dowland, J., & Watson, P. (2021). Run-time adaptation of stream processing spanning the cloud and the edge. In Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion, 1–7.
11. Mattaparthi, R. (2025). GenAI-Augmented Diagnostic Reasoning for Diesel Engine Fault Triage: A Large Language Model Framework for Technician Decision Support at Scale. Journal of Material Sciences & Manufacturing Research, 6(12), 1.
12. Svoboda, J., & Minner, S. (2022). Tailoring inventory classification to industry applications: The benefits of understandable machine learning. International Journal of Production Research, 60(1), 388–401.
13. Mediavilla, M., Dietrich, F., & Palm, D. (2022). Review and analysis of artificial intelligence methods for demand forecasting in supply chain management. Procedia CIRP, 107, 1126–1131.
14. Kolla, S. K., & Reddy, V. A. R. (2024). Evaluating Cloud-Native vs. Hybrid Architectures for Health Benefit Administration Systems. International Journal of Medical Toxicology and Legal Medicine, 27(5), 1042-1053.
15. Zohdi, M., Rafiee, M., Kayvanfar, V., & Salamiraad, A. (2022). Demand forecasting based machine learning algorithms on customer information: An applied approach. International Journal of Information Technology, 14(4), 1937–1947.
16. Wang, H., Xie, F., & Duan, Q. (2022). Federated learning for supply chain demand forecasting. Mathematical Problems in Engineering, 2022, Article 4109070.
17. Kalisetty, S., & Inala, R. (2025). Designing Scalable Data Product Architectures With Agentic AI And ML: A Cross-Industry Study Of Cloud-Enabled Intelligence In Supply Chain, Insurance, Retail, Manufacturing, And Financial Services. Metallurgical and Materials Engineering, 86-98.

P-ISSN: 3050-9734 E-ISSN: 3050-9742
EUROPEAN ADVANCED JOURNAL FOR EMERGING TECHNOLOGIES
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 08 REVISED: JANUARY 24 ACCEPTED: FEBRUARY 05 PUBLISHED: FEBRUARY 18

18. Punia, S., & Shankar, S. (2022). Predictive analytics for demand forecasting: A deep learning-based decision support system. Knowledge-Based Systems, 258, 109956.
19. Loganathan, R. (2025). AGENTIC AI FRAMEWORKS FOR AUTONOMOUS RISK DETECTION AND COMPLIANCE REMEDIATION IN ENTERPRISE DATA CENTER OPERATIONS. Lex Localis-Journal of Local Self-Government, 23 (S6), 9672–9697.
20. Calavaro, C., Russo Russo, G., & Cardellini, V. (2022). Real-time analysis of market data leveraging Apache Flink. In Proceedings of the 16th ACM International Conference on Distributed and Event-Based Systems, 162–165.
21. Verwiebe, J., Grulich, P. M., Traub, J., & Markl, V. (2022). Algorithms for windowed aggregations and joins on distributed stream processing systems. Datenbank-Spektrum, 22, 99–107.
22. Mangalampalli, B. M., Kolla, S. K., Bandi, V. D. V. K., Yandamuri, U. S., & Rani, P. S. (2025). Designing Intelligent Healthcare Ecosystems through Adaptive Data Integration and Autonomous Learning Systems. Vascular and Endovascular Review, 8(20s), 330-347.
23. Agnihotri, P., Koldehofe, B., Binnig, C., & Luthra, M. (2022). PANDA: Performance prediction for parallel and dynamic stream processing. In Proceedings of the 16th ACM International Conference on Distributed and Event-Based Systems.
24. Salari, N., Liu, S., & Shen, Z.-J. M. (2022). Real-time delivery time forecasting and promising in online retailing: When will your package arrive? Manufacturing & Service Operations Management, 24(3), 1421–1436.
25. Mahadevan, S. (2025). DRIVING SUSTAINABILITY IN AUTO & MANUFACTURING SECTORS: THE ROLE OF CLOUD-BASED SERVERLESS AUTOMATION FOR ERP SYSTEMS. International Journal of Applied Mathematics, 38(7s), 1813-1825.
26. van der Hagen, L., Agatz, N., Spliet, R., Visser, T. R., & Kok, L. (2022). Machine learning–based feasibility checks for dynamic time slot management. Transportation Science.
27. Liu, Z., Li, X., & Khojandi, A. (2022). The flying sidekick traveling salesman problem with stochastic travel time: A reinforcement learning approach. Transportation Research Part E: Logistics and Transportation Review, 164, 102816.
28. Mahadevan, S. (2024). Intelligent Serverless Process Automation for Scalable Cloud Operations and Dynamic Resource Optimization. Journal of Computational Analysis and Applications (JoCAAA), 33(06), 4207-4221.
29. Aldunate, Á., Maldonado, S., Vairetti, C., & Armelini, G. (2022). Understanding customer satisfaction via deep learning and natural language processing. Expert Systems with Applications, 209, 118309.
30. Tiwari, A., Khandwe, A., Saha, S., Ramnani, R., Maitra, A., & Sengupta, S. (2023). Towards personalized persuasive dialogue generation for adversarial task oriented dialogue setting. Expert Systems with Applications, 213, 118775.
31. Mangala, N., & Kolla, S. H. (2025). Real-Time Feature Engineering for Streaming AI Workloads Using PySpark and Azure Event Hubs. International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management, 1(6), 93-105.
32. Zhu, Y., Zhang, J., & Liang, J. (2023). Concrete or abstract: How chatbot response styles influence customer satisfaction. Electronic Commerce Research and Applications, 62, 101317.
33. Silva, F. A., Shojaei, A. S., & Barbosa, B. (2023). Chatbot-based services: A study on customers' reuse intention. Journal of Theoretical and Applied Electronic Commerce Research, 18(1), 457–474.
34. Nagubandi, A. R. (2024). Breakthrough Real-Time AI-Driven Regulatory Intelligence for Multi-Counterparty Derivatives and Collateral Platforms: Autonomous Compliance for IFRS, EMIR, NAIC, SOX & Emerging Regulations. Journal of Information Systems Engineering and Management, 9(4).
35. Özarık, S. S., da Costa, P., & Florio, A. M. (2023). Machine learning for data-driven last-mile delivery optimization. Transportation Science, 58(1), 27–44.
36. Dieter, P., Caron, M., & Schryen, G. (2023). Integrating driver behavior into last-mile delivery routing: Combining machine learning and optimization in a hybrid decision support framework. European Journal of Operational Research, 311(1), 283–300.
37. Tian, Y.-X., & Zhang, C. (2023). An end-to-end deep learning model for solving data-driven newsvendor problem with accessibility to textual review data. International Journal of Production Economics, 265, 109016.





38. Mangalampalli, B. M., & Kolla, S. K. (2025). Large Language Models for Automated Healthcare Data Dictionary Generation and Maintenance. Vascular and Endovascular Review, 8(20s), 363-375.
39. Yang, C.-H., Wang, H.-T., Ma, X., & Talluri, S. (2023). A data-driven newsvendor problem: A high-dimensional and mixed-frequency method. International Journal of Production Economics, 266, 109042.
40. Park, H., Choi, D. G., & Min, D. (2023). Adaptive inventory replenishment using structured reinforcement learning by exploiting a policy structure. International Journal of Production Economics, 266, 109029.
41. Ramírez-Villamil, A., Montoya-Torres, J. R., Jaegler, A., & Cuevas-Torres, J. M. (2023). Reconfiguration of last-mile supply chain for parcel delivery using machine learning and routing optimization. Computers & Industrial Engineering, 184, 109604.
42. Bhattacharya, S., Govindan, K., Ghosh Dastidar, S., & Sharma, P. (2024). Applications of artificial intelligence in closed-loop supply chains: Systematic literature review and future research agenda. Transportation Research Part E: Logistics and Transportation Review, 184, 103455.
43. Iseri, F., Iseri, H., Chrisandina, N. J., Iakovou, E., & Pistikopoulos, E. N. (2025). AI-based predictive analytics for enhancing data-driven supply chain optimization. Journal of Global Optimization.