Street Art & Urban Creativity | Vol. 11, No. 7s, 2025 | ISSN 2183-9956
Street Art & Urban Creativity. Scientific Journal

Real-Time Streaming Analytics Architectures for Industrial IoT Applications

GAURAV KUMAR SINHA[1]
[1]7319 LONGSPUR HOLLOW LANE KATY, TX 77493-4704, UNITED STATES

Abstract

Modern factories, warehouses and process plants generate continuous streams of data from machines, tags, meters, cameras and people. The value of much of this data decays within seconds. A vibration anomaly that is detected after a minute may already have damaged a bearing, and a forklift that approaches a worker is a hazard only for as long as the two are close. Industrial Internet of Things applications therefore need streaming analytics architectures that ingest, process and act on data in real time, while coping with unreliable wireless links, heterogeneous protocols, strict safety requirements and very large data volumes. This paper examines such architectures. We describe a layered reference architecture that spans field devices, gateways, a streaming backbone, stateful stream processing, storage and serving, and we compare three architectural patterns: cloud-centric streaming, micro-batch processing and an edge-cloud hybrid. We discuss the time semantics, delivery guarantees, state management and backpressure mechanisms that determine correctness, and we relate the architecture to ten industrial application areas, including maintenance, quality control, localization and safety. In a simulated plant with 2,000 sensors producing about 250,000 events per second, the edge-cloud hybrid produced alerts with a median latency of 38 ms, against 420 ms for cloud-centric streaming and about 5.8 seconds for micro-batching, reduced uplink traffic by 92%, kept safety alerts running during a ten-minute network outage and sustained 900,000 raw events per second. Anomaly detection quality was within one point of the cloud-centric design. The study offers practical guidance for selecting and configuring streaming architectures for industrial settings.

Keywords: Industrial IoT, Stream Processing, Real-Time Analytics, Edge Computing, Event-Time Processing, Predictive Maintenance, Indoor Localization, Smart Manufacturing.

1. Introduction

Industrial operations have always depended on information, but the amount of information available to them has changed in kind. A modern production line carries thousands of sensors that record temperature, pressure, vibration, current and position, and its machines report their states through controllers. A warehouse tracks its pallets, trucks and people with radio tags and cameras. A power or process plant meters its energy and its emissions continuously. The idea of connecting these sources to computing systems, often called the Industrial Internet of Things, is a cornerstone of the broader movement toward smart manufacturing and Industry 4.0 [1], [2]. The promise is that better information produces better decisions: fewer breakdowns, higher quality, lower energy use, safer workplaces and more efficient logistics [3].

To realize this promise, the information must reach the decision at the right moment. Many industrial decisions are time-critical. A warning about a failing bearing is useful only if it arrives early enough for maintenance to be planned, an inspection result must arrive before the defective part moves on, and a proximity alert between a forklift and a pedestrian must arrive in a fraction of a second. Traditional data warehouses and batch analytics, which process data hours or days after it was produced, are suited to reporting and long-term analysis but not to such needs. Streaming analytics processes data continuously as it arrives and produces results with latencies of milliseconds to seconds.

Designing a streaming architecture for an industrial setting is more demanding than for a typical web application. Data sources speak many protocols and operate on unreliable and

sometimes congested wireless links. Data arrive out of order, are delayed or are lost, and the meaning of a result often depends on when an event happened, not on when it was received. Volumes are high, since a single vibration sensor sampled at a thousand times per second produces more data than many business systems do in a day. The plant must keep functioning when the connection to the cloud is interrupted. Safety-related functions have latency and reliability requirements that cannot be left to a distant data center. And the systems must be secure, observable and maintainable by operations staff, not only by software specialists.

This paper addresses these challenges through a systematic treatment of streaming architectures for industrial applications. Its contributions are as follows: (i) a layered reference architecture for industrial streaming analytics, with a mapping to ten application areas; (ii) a comparison of three architectural patterns, cloud-centric streaming, micro-batch processing and an edge-cloud hybrid; (iii) an analysis of the design decisions that determine correctness, namely time semantics, delivery guarantees, state management and flow control; (iv) a simulation study of latency, throughput, bandwidth, accuracy, fault tolerance and scalability; and (v) practical guidance for architects. The remainder of the paper is organized as follows. Section II reviews background, Section III discusses related work, Section IV presents the methodology, Section V describes the experimental setup, Section VI reports results, Section VII discusses implications and limitations, and Section VIII concludes.

2. LITERATURE REVIEW

A. Industrial IoT and Application Areas

Industry 4.0 describes the integration of cyber-physical systems, the Internet of Things and data-driven services into production [1]. Reviews of the field identify smart, connected products and machines, vertical and horizontal integration across the value chain and the use of data analytics as defining elements [2]. The Industrial Internet of Things differs from consumer Internet of Things in its stringent requirements for reliability, determinism, longevity of equipment and security, and in the fact that its systems are coupled to physical processes that can cause harm when they fail [3], [14]. A survey of the Internet of Things broadly covers architectures, enabling technologies, security and applications and shows that industrial use is among the most demanding [22].

Fig. 1 summarizes the main application areas of industrial analytics around a production line. The figure arranges ten areas around a line with robots, conveyors, a machine tool and a logistics zone: monitoring, the human-machine interface, product design, maintenance, energy optimization, localization, manufacturing optimization, quality control, logistics, and safety and security. Four of these, maintenance, localization, quality control, and safety and security, are set in bold in the figure, which marks them as particularly central. The areas differ in what they ask of an analytics architecture. Maintenance needs continuous analysis of high-rate signals to detect degradation, quality control needs inspection results while a part is still on the line, localization needs frequent position estimates from radio measurements, and safety needs the fastest and most reliable alerts of all. Others, such as energy optimization and logistics, tolerate delays of seconds or minutes and benefit more from large-scale aggregation.

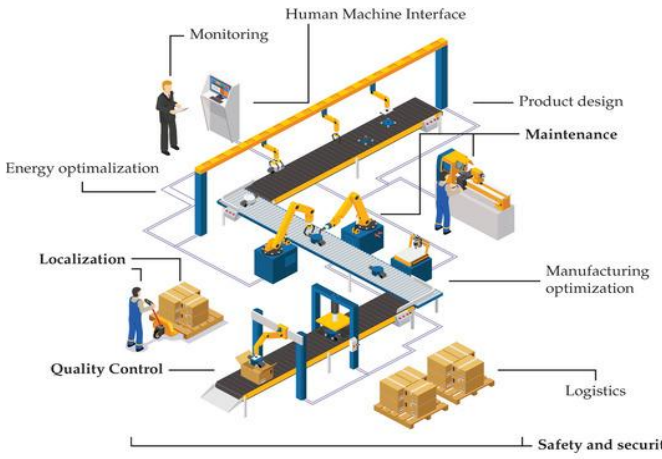


Fig. 1. Application areas of industrial analytics around a production line, including maintenance, localization, quality control, and safety and security.

B. Stream Processing Foundations

Stream processing systems treat data as an unbounded sequence of events and compute continuously over it. Early designs used the micro-batch approach, in which the stream is divided into short intervals and each interval is processed as a small batch, which provides strong fault tolerance by recomputation at the cost of latency bounded below by the interval length [8]. Later engines process events one at a time with managed state, achieving much lower latency, and unify batch and stream processing in a single model [7]. A influential formulation of this model, the dataflow model, separates four questions: what is computed, where in event time the results are grouped, when in processing time the results are emitted and how later refinements relate to earlier results [6]. It introduced windows over event time, watermarks that estimate how complete the input is, and triggers that decide when to emit results, concepts that are essential for industrial data in which events are delayed or reordered.

State and fault tolerance are central concerns. Stream processors maintain state, such as the running statistics of a sensor or the recent history of a machine, and they must recover it after failures without losing or duplicating results. Consistent distributed snapshots, in which the state of a distributed computation is captured without stopping it, are the basis of the checkpointing mechanisms used in modern engines [10], [11]. Delivery guarantees are classified as at-most-once, at-least-once and exactly-once effects, and achieving exactly-once effects requires cooperation between the broker, the processor and the sink. A distributed log, which stores events durably in ordered partitions that consumers read independently, has become the standard backbone for ingestion because it decouples producers from consumers, buffers bursts and allows replay [9]. Recent surveys trace how stream processing systems have evolved in architecture, state management and scalability [24].

C. Architectural Patterns

Two patterns have shaped discussion. The lambda architecture maintains two paths, a batch layer that recomputes accurate results over all data and a speed layer that provides low-latency approximate results, with a serving layer that merges them [4]. It offers robustness but requires two code bases to be kept consistent, which is a significant maintenance burden. The kappa architecture simplifies this by using a single streaming path over a replayable log, with reprocessing performed by replaying the log through a new version of the logic [5], [25]. In industrial settings, a third consideration arises, the location of computation. Edge computing moves processing close to the data source to reduce latency and bandwidth and to keep working during disconnection [12], [13]. The combination of edge preprocessing with cloud-scale analytics gives rise to hybrid architectures, which we study in this paper.

D. Industrial Connectivity and Localization

Industrial devices communicate through a variety of protocols. Field-level controllers exchange data through fieldbus and industrial Ethernet systems, and the discussion of the future of industrial communication emphasizes convergence on Ethernet-based networks and wireless technologies while maintaining determinism where needed [14]. Interoperability standards such as the OPC Unified Architecture provide a platform-independent information model for industrial data [15], and lightweight publish-subscribe messaging protocols such as MQTT are widely used to carry telemetry from constrained devices [16]. Wireless sensor networks are increasingly used in factories and warehouses where wiring is impractical. A common application is indoor localization, in which the position of a mobile asset or person is estimated from radio measurements between a mobile node and fixed reference nodes whose positions are known, since satellite positioning does not work well indoors [19]. Fig. 3, discussed in Section IV, shows such a deployment.

E. Analytics, Anomalies and Drift

Analytics on industrial streams include anomaly detection, which flags unusual behavior without needing labeled failures [20], prediction of remaining useful life and failures [18], and classification of quality events. A recurring difficulty is concept drift, in which the statistical behavior of a process changes over time because of wear, maintenance, new products or seasonal conditions, so that models trained on past data become less accurate [21]. A streaming architecture therefore needs to support not only the execution of models but also their monitoring and updating, a point we return to in the methodology.

3. RELATED WORK

Research relevant to this paper can be grouped into industrial architectures, stream processing systems, edge and fog computing for industry, and localization in industrial environments. Industrial architectures proposed for cyber-physical production describe layered structures that connect machines, networks, analytics and decision-making, usually in five levels from sensing to cognition [17]. Such frameworks provide conceptual guidance but do not generally specify how streaming engines, brokers and edge nodes should be combined, nor do they quantify the consequences of design choices for latency, bandwidth and resilience.

Stream processing research provides the engines and the theory. The dataflow model offers a principled treatment of time, windows and completeness [6]. Engines that combine stateful per-event processing with consistent snapshots demonstrate that low latency and strong guarantees can coexist [7], [10]. Distributed logs provide the durable, partitioned transport on which these engines depend [9]. Much of this work is evaluated on web-scale workloads such as clickstreams and logs, which differ from industrial workloads in several respects. Industrial data come from devices with limited connectivity, often at very high sample rates, with strong requirements for operation during disconnection and with consequences in the physical world when delays occur.

Edge and fog computing research argues that processing close to the source is necessary for latency-sensitive and bandwidth-intensive applications and that the cloud and the edge should be viewed as a continuum [12], [13]. Studies in industrial settings apply edge nodes to filter, aggregate and analyze data before sending them upstream, and they report reductions in bandwidth and latency. Many of these studies focus on one application or one technique and do not compare complete architectural patterns under common conditions, which is the gap this paper aims to fill.

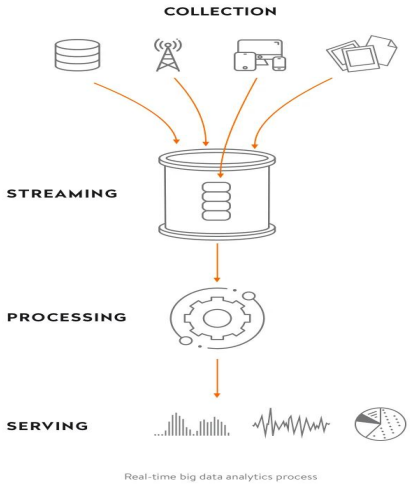
Localization research surveys technologies for indoor positioning, from radio signal strength to time-of-flight methods, and discusses their accuracy, cost and deployment complexity [19]. In industrial environments, metal structures, moving vehicles and people create strong multipath effects that degrade measurements, which makes filtering and fusion of the measurements important. The processing of localization data as a stream, with estimates updated continuously and combined with other signals, is a natural fit for streaming architectures but is seldom discussed together with the architecture itself.

Three open issues remain. First, there is a lack of comparative evidence on how cloud-centric, micro-batch and hybrid patterns behave under industrial conditions. Second, correctness issues such as event-time handling, delivery guarantees and backpressure are rarely examined in industrial scenarios. Third, the relation between the architecture and the full range of application areas, from maintenance to safety, is not systematically analyzed. This paper addresses these issues.

4. METHODOLOGY

A. Reference Architecture

The reference architecture follows the four-stage process illustrated in Fig. 2, which depicts real-time big data analytics as collection, streaming, processing and serving. At the top, collection gathers data from heterogeneous sources, shown in the figure as databases, wireless infrastructure, user devices and documents or images. The collected data flow into a streaming stage, drawn as a container holding a stack of records, which represents a durable broker that buffers and orders events. A processing stage, drawn as a gear, applies computation to the streams. At the bottom, a serving stage presents results as charts and signals, such as histograms, time series and scatter or pie charts, for people and systems that act on them.



.

Fig. 2. Real-time big data analytics process: collection, streaming, processing and serving.

We refine this process into seven layers for industrial use, summarized in Table I. The field layer contains sensors, actuators, controllers and tags. The edge layer contains gateways and edge servers that sit within the plant. The ingestion layer is the broker, which accepts events from the field and edge layers. The processing layer runs stateful stream computations. The storage layer retains data in tiers, with a hot tier for recent data that is needed at low latency, a warm tier for weeks of history and a cold tier for long-term archives. The serving layer exposes results through dashboards, alerts and interfaces to other systems, including the controllers that close the loop to the process. A governance layer cuts across the others, handling security, schema management, monitoring and the lifecycle of models.

TABLE I. Layers of the Industrial Streaming Reference Architecture

Layer	Role	Typical Components
Field	Sense, actuate and identify	Sensors, controllers, tags, cameras
Edge	Connect, filter and act locally	Gateways, edge servers, local analytics
Ingestion	Buffer, order and distribute events	Partitioned durable log, protocol adapters
Processing	Compute continuously with state	Stream engine, windows, pattern detection, model scoring
Storage	Retain data in tiers	Time-series store, object storage, feature store
Serving	Deliver results and actions	Dashboards, alerting, application interfaces, controller feedback
Governance	Secure, observe and manage	Identity, schema registry, monitoring, model lifecycle

B. Mapping Application Areas to Requirements

The ten application areas of Fig. 1 place different demands on the architecture. Table II characterizes them by their typical data and by the latency within which a result is useful. Safety and security have the tightest budgets, on the order of tens of milliseconds, and must keep working when the network to the cloud fails. Quality control and localization need results within a few hundred milliseconds. Maintenance has a range of needs, from near-real-time detection of severe anomalies to slower trend analysis. Energy optimization, logistics and manufacturing optimization are less time-critical and benefit from the larger context that cloud-scale analytics can provide. These differences motivate a design in which each analysis runs at the layer that best matches its latency and scope.

TABLE II. Application Areas and Their Requirements on the Streaming Architecture

Application Area	Typical Data	Useful Latency	Preferred Location
Safety and security	Proximity, position and access events	Under 100 ms	Edge
Quality control	Inspection results, process parameters	50 to 500 ms	Edge, with cloud learning
Localization	Radio signal measurements, ranges	100 to 1,000 ms	Edge
Maintenance	Vibration, temperature, current	100 ms to minutes	Edge detection, cloud trending
Manufacturing optimization	Machine states, throughput, cycle times	Seconds	Edge and cloud
Energy optimization	Meter and utilization readings	Seconds to minutes	Cloud
Logistics	Scan, tag and vehicle events	Seconds	Edge and cloud
Monitoring and human-machine interface	Aggregated states and alarms	About 1 second	Edge and cloud

C. Architectural Patterns Compared

We compare three patterns. In the cloud-centric streaming pattern, all raw events are sent from the plant to a cloud broker and processed there by a stream engine, and results return to the plant. It is simple to operate and offers elastic scale, but it consumes a great deal of bandwidth and depends on the uplink for every function. In the micro-batch pattern, events are collected and processed in short batches, here of ten seconds, which simplifies fault tolerance and allows efficient processing, but sets a lower bound on latency equal to the batch interval [8]. In the edge-cloud hybrid pattern, edge servers in the plant ingest raw events, perform filtering, feature extraction, local model scoring and time-critical alerting, and forward compact summaries and selected raw segments to the cloud, which performs plant-wide and cross-site analytics and trains models. The hybrid occupies a position between the lambda and kappa designs, with a single streaming logic that runs at both levels, so that it avoids maintaining separate batch and speed implementations [4], [5].

D. Edge Layer and Wireless Deployment

Fig. 3 shows the physical setting of the edge layer in an industrial facility that combines a warehouse and a production building. An office houses a process automation controller. A gateway connects the wireless network in the facility to a network controller, which manages the nodes. Throughout the building, two kinds of wireless nodes are deployed. Reference nodes are installed at fixed, known positions, and sensor nodes are distributed over the facility on assets, vehicles and structures, and the radio measurements between them make it possible to estimate the positions of mobile nodes. The facility contains trucks, forklifts, shelves and people, which move through the space and create the radio reflections and obstructions that make indoor localization challenging.

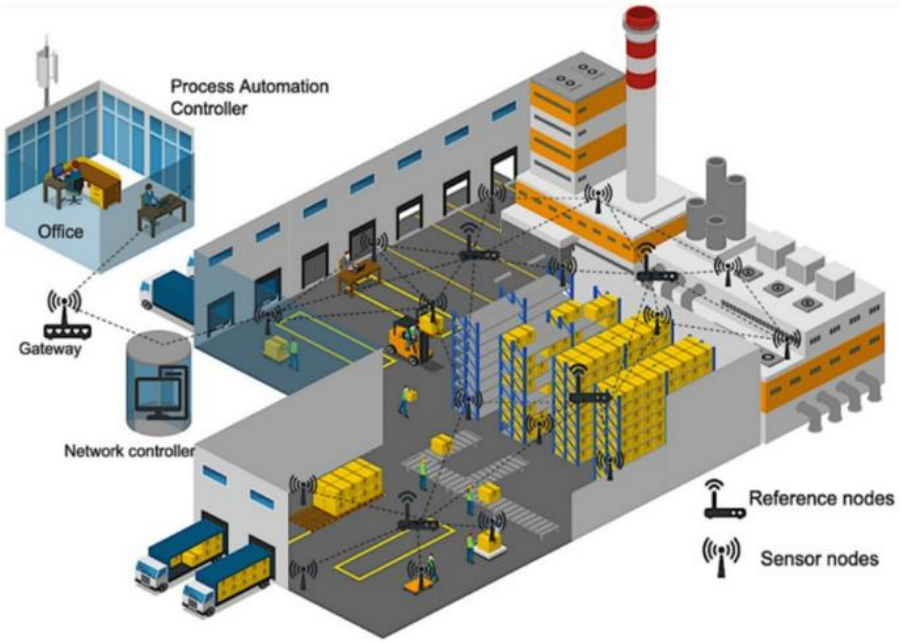


Fig. 3. Wireless deployment in an industrial facility: process automation controller, gateway, network controller, fixed reference nodes and sensor nodes.

In the hybrid architecture, the gateway and the edge server perform three tasks for such a deployment. They ingest raw measurements from the nodes and normalize them into a common event format with timestamps and device identifiers. They filter and smooth measurements, which reduces the effect of multipath and noise, and they estimate positions with a filter that fuses the measurements over time. They also evaluate safety rules locally, such as the proximity of a forklift to a pedestrian, and raise alerts without any round trip to the cloud. The position streams and alerts are then published to the broker, and compact summaries are forwarded upstream.

E. Time Semantics and Delivery Guarantees

Industrial streams are rarely well ordered. Wireless links delay and reorder messages, gateways batch transmissions, and devices occasionally resend. The architecture therefore uses event time, the time at which a measurement was taken as recorded by the device or the gateway, for all windowed computations, and it uses watermarks to estimate how far the stream has progressed. A configurable allowed lateness defines how long after the watermark a late event is still accepted into a window, with later events routed to a side output for correction. Using event time prevents the false alarms and missed detections that occur when delays are mistaken for changes in the process. Delivery guarantees are set according to the use. Safety alerts and billing-related counts use effects that occur exactly once, which are achieved by transactional writes and checkpointed state, at the cost of some additional latency. High-rate telemetry for trending uses at-least-once delivery with idempotent sinks, which is cheaper and tolerates occasional duplicates.

F. State, Fault Tolerance and Flow Control

Stream computations keep state, such as the running baseline of a machine's vibration or the recent positions of a vehicle. State is checkpointed periodically to durable storage using consistent snapshots, so that after a failure the computation resumes from the last checkpoint and replays the log from the corresponding position [10], [11]. At the edge, the broker and the state are stored on local disks and replicated across at least two edge servers where safety functions are involved. When the uplink fails, the edge continues to operate and buffers data for the cloud, and when the connection returns, buffered data are replayed in order with rate limits so as not to overwhelm the cloud. Flow control, or backpressure, ensures that a slow consumer slows its producers instead of exhausting memory. The architecture applies backpressure from processing to ingestion and, where needed, sheds load in a controlled way, dropping low-priority telemetry before safety-critical events.

G. Analytics and Model Lifecycle

Models are trained in the cloud on historical data, and packaged for deployment to the edge, where they score the stream. Examples include anomaly detectors for vibration, classifiers for inspection results and filters for localization. Models are monitored for drift by comparing the distribution of inputs and of predictions with their reference, and a drift alarm triggers retraining or review [21]. New versions are deployed gradually, first in shadow mode, where they run alongside the current model without affecting actions, and then in a canary stage on a subset of machines. This design keeps the logic consistent across the edge and the cloud, since the same feature definitions are used in both.

H. Evaluation Metrics

We report alert latency, measured from the moment an event occurs to the moment an alert is issued, using median and tail percentiles, the maximum sustained rate of raw events that an architecture handles without growing queues, and the uplink bandwidth consumed. Accuracy is reported for each workload with the measure appropriate to it: the F-score for vibration anomalies, recall for quality events, mean position error in meters for localization, and the rate of missed proximity warnings for safety. Resilience is measured by the loss of alerts and the time to recover during failures. Scalability is measured by latency and the number of processing nodes as the event rate grows. Duplicates and false alarms are counted separately.

5. EXPERIMENTAL SETUP

A. Simulated Plant

The evaluation used a simulated plant modeled on the setting of Fig. 3, with a production hall and a warehouse. The plant contained 2,000 sensors of four kinds. A group of 200 vibration sensors on rotating machines sampled at one kilohertz and produced the largest volume of data. A group of quality stations produced inspection events at rates determined by the line speed. An indoor positioning system with 60 reference nodes and 300 sensor nodes attached to forklifts, pallets and workers produced position measurements several times per second. A group of energy and process meters produced readings every second. Together, the plant produced about 250,000 raw events per second, with an average size of about 48 bytes, equivalent to roughly 96 megabits per second of raw traffic.

B. Architectures Under Test

The three architectures were implemented in the simulation with the same logical analytics. The cloud-centric configuration forwarded all raw events through a gateway to a cloud broker and a cloud stream engine. The micro-batch configuration collected events in ten-second batches, compressed them and processed them in the cloud. The hybrid configuration ran ingestion, filtering, feature extraction, model scoring and safety rules on edge servers in the plant, and forwarded compact summaries, about twelve thousand events per second, along with selected raw segments around anomalies to the cloud. Cloud processing used a partitioned broker and a stateful stream engine with checkpointing every ten seconds.

C. Network and Failure Conditions

The uplink from the plant had a capacity of 150 megabits per second and a latency of about 30 milliseconds each way, with random jitter and occasional loss. The wireless network within the plant was subject to delays and reordering that were modeled as a random variation of up to two seconds in the arrival of a small share of events. Three fault scenarios were applied: an outage of the uplink for ten minutes, the failure of a broker node and the failure of a stream processing task. Delivery guarantees were varied between at-least-once and exactly-once effects.

D. Workloads and Ground Truth

For vibration anomalies, faults were injected into the signals at known times, with different severities and progression rates, and the detector was scored against these labels. For quality events, a share of units carried injected defects, and recall was measured against them. For localization, the true positions of mobile nodes were known from the simulation, and the position error was computed against them. For safety, encounters between vehicles and pedestrians were generated, and the proximity warnings were scored against the true

encounters. Each scenario was simulated for twenty-four hours of plant time with ten random seeds, and results are means. Differences below the seed-to-seed variation, about three percent for latency percentiles, are not interpreted.

E. Configuration

Table III summarizes the main settings. The streaming logic used event time with a watermark and an allowed lateness of two seconds for time-critical streams and thirty seconds for trending streams. Windows were one second for safety and localization, and ten seconds for maintenance features. The simulation is a model of the processing, network and failure behavior and not a measurement on a physical plant, so the results indicate relative behavior and not absolute performance.

TABLE III. Experimental Configuration

Component	Setting
Plant	2,000 sensors, about 250,000 raw events per second, about 96 Mbit/s raw
Vibration sensors	200 sensors at 1 kHz
Positioning system	60 reference nodes, 300 mobile sensor nodes
Uplink	150 Mbit/s, about 30 ms one way, with jitter and loss
Micro-batch interval	10 seconds
Checkpoint interval	10 seconds
Allowed lateness	2 seconds for time-critical, 30 seconds for trending streams
Fault scenarios	Uplink outage of 10 minutes, broker node failure, task failure
Repetitions	Ten seeds, 24 hours of plant time, means reported

6. RESULTS

A. Latency, Throughput and Bandwidth

Table IV compares the three architectures. Cloud-centric streaming produced alerts with a median latency of 420 ms and a 99th percentile of 2,300 ms, which is acceptable for trending and logistics but too slow for safety. The micro-batch configuration had a median of 5,800 ms and a 99th percentile of 11,600 ms, since results could not appear before the end of each ten-second batch. The hybrid produced edge alerts with a median of 38 ms, a 95th percentile of 85 ms and a 99th percentile of 160 ms. Plant-wide analytics in the cloud under the hybrid had a median of 310 ms, similar to the cloud-centric design, because they used the same cloud path.

TABLE IV. Alert Latency, Throughput and Uplink Traffic by Architecture

Architecture	p50 (ms)	p95 (ms)	p99 (ms)	Max Sustained Raw Events (per second)	Uplink Traffic (Mbit/s)
Cloud-centric streaming	420	1,150	2,300	310,000	96
Micro-batch (10 s)	5,800	10,900	11,600	340,000	52
Edge-cloud hybrid (edge alerts)	**38**	**85**	**160**	**900,000**	**7.7**

The hybrid also sustained the highest raw event rate, about 900,000 events per second, since the edge servers shared the ingestion work, whereas the cloud-centric design saturated its broker partitions at about 310,000 events per second. The uplink traffic of the hybrid, 7.7 megabits per second, was 92% lower than that of cloud-centric streaming, which consumed 96 megabits per second, and 85% lower than that of the compressed micro-batch configuration. This reduction is the product of extracting features at the edge and forwarding summaries, with raw data around anomalies sent only when needed. The reduction also means that the architecture has headroom to grow the number of sensors on the same uplink.

B. Analytic Accuracy

A concern with edge summarization is that it could discard information that the cloud would have used. Table V shows that the loss was small. For vibration anomalies, the F-score was 0.91 for the cloud-centric design and 0.90 for the hybrid. Recall on quality events was 94.1% against 93.6%. Localization was slightly better in the hybrid, with a mean error of 1.7 meters against 1.9, because the edge filter had access to every raw measurement at full rate and fused measurements locally before any delay or loss on the uplink. The miss rate of proximity warnings was 0.9% in the hybrid and 2.8% in the cloud-centric design, since in the latter, delayed alerts often arrived after the encounter had passed. In summary, the hybrid matched the accuracy of the cloud-centric design for analytics and clearly improved the time-critical functions.

TABLE V. Analytic Accuracy by Workload

Workload and Measure	Cloud-Centric Streaming	Edge-Cloud Hybrid
Vibration anomaly detection (F-score)	0.91	0.90
Quality event recall (%)	94.1	93.6
Localization mean error (m)	1.9	1.7
Missed proximity warnings (%)	2.8	0.9

C. Fault Tolerance

Table VI summarizes the fault scenarios. During a ten-minute uplink outage, the cloud-centric design lost all real-time analytics and alerts for the duration, and some events were lost when the gateway buffer overflowed. The hybrid continued to generate edge alerts without interruption, buffered the data locally and, when the connection returned, replayed them in order within about 3.5 minutes with rate limits. A broker node failure was recovered in about 24 seconds in both cases, thanks to replication and the ability of consumers to resume from their stored positions. A failure of a processing task was recovered from the last checkpoint in about 12 seconds in the hybrid, and replay from the log restored the state without loss. Delivery guarantees had a clear effect. With at-least-once delivery, duplicates accounted for about 0.8% of events after failures, which would be harmless for trending but unacceptable for counts. With exactly-once effects, no duplicates occurred, at a cost of about 14% higher median latency.

TABLE VI. Behavior Under Failures and Delivery Guarantees

Scenario	Cloud-Centric Streaming	Edge-Cloud Hybrid
Uplink outage of 10 minutes: real-time alerts	Lost for the whole outage	Continuous, no alerts lost
Uplink outage: data recovery after reconnection	Partial, buffer overflow at gateway	Complete, replayed in about 3.5 minutes
Broker node failure: recovery time	About 24 s	About 24 s
Task failure: recovery from checkpoint	About 18 s	About 12 s
At-least-once delivery: duplicate events	About 0.8%	About 0.8%
Exactly-once effects: duplicates and latency cost	None, about 14% higher latency	None, about 14% higher latency

D. Scalability

Table VII shows how the hybrid scaled with event rate. At 50,000 events per second, two processing nodes were enough and the 95th-percentile alert latency was 70 ms. At 200,000 events per second, six nodes were needed and latency was 82 ms, at 500,000 events per second

fourteen nodes and 105 ms, and at one million events per second twenty-eight nodes and 160 ms. The number of nodes grew roughly in proportion to the rate, and latency increased modestly, which indicates near-linear scaling. Beyond about 900,000 raw events per second, the plant's edge ingestion began to saturate, and the extra capacity at one million events per second came from scaling out the edge servers.

TABLE VII. Scalability of the Edge-Cloud Hybrid

Raw Events per Second	Processing Nodes	p95 Alert Latency (ms)
50,000	2	70
200,000	6	82
500,000	14	105
1,000,000	28	160

E. Design Choice Ablation

Table VIII shows the effect of removing individual design elements from the hybrid. Replacing event time with processing time, which treats the arrival time as the time of the event, lowered the F-score for vibration anomalies from 0.90 to 0.84 and raised false alarms from about 3.1 to 7.9 per day, because delayed and reordered messages were mistaken for changes in the machine. Removing edge filtering restored the uplink traffic to the full 96 megabits per second. Disabling backpressure caused queues to build during bursts, so that the 99th-percentile alert latency rose to 1,900 ms and about 2.1% of events were dropped. Disabling checkpointing meant that recovery after a task failure required rebuilding state from scratch, which took about 6.5 minutes. Each of these elements therefore had a clear and independent contribution.

TABLE VIII. Ablation of Design Elements in the Edge-Cloud Hybrid

Variant	Effect Observed
Full design	F-score 0.90, about 3.1 false alarms per day, p99 alert latency 160 ms, uplink 7.7 Mbit/s
Processing time instead of event time	F-score 0.84, about 7.9 false alarms per day
No edge filtering and summarization	Uplink traffic 96 Mbit/s
No backpressure control	p99 alert latency 1,900 ms, about 2.1% of events dropped in bursts
No state checkpointing	Recovery after task failure about 6.5 minutes

F. Summary by Application Area

The results can be related to the application areas of Fig. 1. For safety and security, the hybrid was the only architecture that met a latency budget of under 100 ms with a 95th percentile of 85 ms, and the only one that kept working during an outage. For localization and quality control, the hybrid and cloud-centric designs could both meet budgets of a few hundred milliseconds, although the hybrid was more resilient and more accurate for localization. For maintenance, edge detection allowed rapid response to severe anomalies, and the forwarding of raw data around those anomalies gave the cloud the evidence it needed for model training. For energy optimization and logistics, the cloud path was sufficient, and the micro-batch pattern was a reasonable and cheap choice when delays of seconds could be tolerated.

7. DISCUSSION

The results show that no single architecture is best for every industrial analytic. Cloud-centric streaming is simple and elastic, but it is dependent on the uplink, consumes large bandwidth and cannot meet the latency demands of safety functions. Micro-batching is cheap and robust, but its latency cannot be lower than the batch interval, which excludes it from interactive and time-critical uses. The edge-cloud hybrid meets the hardest requirements, reduces bandwidth by an order of magnitude and continues to operate through outages, at the price of greater

architectural complexity, since computation is distributed over more locations and models must be deployed and monitored in the plant.

The design decisions analyzed in the methodology proved to be as important as the choice of pattern. Event-time processing with watermarks avoided a substantial loss of accuracy that arose when delays on wireless links were mistaken for changes in the process. Checkpointing and replay limited the cost of failures. Backpressure and prioritized load shedding protected safety-critical events during bursts. The choice of delivery guarantee reflected a trade-off between duplicates and latency that should be made per stream and not for the whole system. These mechanisms are not optional refinements. They determine whether an architecture produces trustworthy results under real industrial conditions.

The deployment of Fig. 3 illustrates why the edge matters for localization and safety. Radio measurements in a facility full of metal, vehicles and people are noisy, and the quality of a position estimate depends on filtering and fusing measurements at their full rate, which is best done near the source. The same edge servers can evaluate proximity rules and alert immediately, without waiting for the cloud. At the same time, the cloud retains an essential role in which scale and context matter, including cross-site comparison, long-term trending, model training and optimization of logistics and energy use across the enterprise. The architecture therefore divides work according to latency and scope, as Table II suggests.

Several limitations should be noted. The study is based on simulation of a single plant, and real plants exhibit greater diversity of devices, protocols, network conditions and operational practices. Wireless propagation and device behavior were modeled in a simplified way. The costs and effort of operating edge infrastructure, including upgrades, security patches and physical maintenance, were not included in the comparison, and they may reduce the advantage of the hybrid for small sites. The accuracy results depend on the injected faults and defects, which may differ from those in practice. Security, which is essential in industrial settings, was addressed only at the level of architecture, with the governance layer, and was not evaluated quantitatively. Standards for industrial security, for example, impose requirements on segmentation, access control and patching that affect how streaming systems are deployed [26]. Finally, safety functions that can cause physical harm should not rely solely on analytics pipelines of this kind and must be backed by certified safety systems.

Practical guidance follows from these findings. Classify each analytic by the latency within which its result is useful and by whether it must survive loss of connectivity, and place it accordingly, with time-critical and resilient functions at the edge. Use event time and watermarks for all windowed computations. Choose delivery guarantees per stream. Apply backpressure and prioritize critical events when shedding load. Summarize at the edge, and forward raw data selectively around anomalies. Use a single streaming logic at the edge and in the cloud to avoid maintaining two code bases, and monitor models for drift. Finally, treat governance, including security, schema management and the model lifecycle, as part of the architecture from the start.

8. CONCLUSION

This paper examined real-time streaming analytics architectures for industrial Internet of Things applications. We presented a seven-layer reference architecture based on the four stages of collection, streaming, processing and serving, related it to ten industrial application areas, compared cloud-centric, micro-batch and edge-cloud hybrid patterns, and analyzed the time semantics, delivery guarantees, state management and flow control that determine correctness. In a simulated plant with 2,000 sensors and about 250,000 events per second, the hybrid produced alerts with a median latency of 38 ms, against 420 ms for cloud-centric streaming and 5.8 seconds for micro-batching, reduced uplink traffic by 92%, sustained 900,000 raw events per second, kept alerts running through a ten-minute outage and matched the analytic accuracy of the cloud-centric design within about one point. Ablation showed that event-time processing, edge summarization, backpressure and checkpointing each made a distinct contribution.

Future work will proceed in four directions. The first is validation in real plants, with physical devices, real networks and operating staff. The second is the integration of learning at the edge, including federated and incremental training, so that models adapt to local conditions without moving raw data. The third is a quantitative treatment of security and of the cost of operating edge infrastructure, including an analysis of the total cost of ownership. The fourth is the integration of streaming analytics with digital twins [23] and with closed-loop control, in a way

that preserves the safety guarantees of the underlying automation.

References

1. Pawan Kumar Singh, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5713
2. Pawan Kumar Singh, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5712
3. A. Mahendra Vardhan and S. Sridhar, "Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 622-625, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458
4. Ronakkumar shah , Cognitive Process Intelligence: Integrating Generative AI and Automated Workflow Discovery, Vol. 33 No. 08 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5676
5. Ronakkumar shah, research Trends in Smart Workflow Automation: A Bibliometric Analysis of Emerging Themes, Home / Archives / Vol. 33 No. 05 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5675
6. Dip Bharatbhai Pate, Dynamic vs. Static visualizations: Understanding their use cases in big data analysis, International Journal of Science and Research Archive, 2022, 05(02), 392-395., DOI https://doi.org/10.30574/ijsra.2022.5.2.0077
7. Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: https://doi.org/10.30574/ijsra.2021.2.1.0047
8. Dip Bharatbhai Patel, Understanding the Impact of Text Analytics on Social Media Sentiment Analysis, Vol. 5 No. 5 (2019) ,September-October, DOI: https://doi.org/10.32628/CSEIT1936413
9. Dip Bharatbhai Patel, Real-Time Data Streaming in BI Dashboards: Opportunities and Challenges, Archives / Vol. 5 No. 4 (2019) : July-August, DOI: https://doi.org/10.32628/CSEIT192142
10. Dip Bharatbhai Patel, the Role of Data Visualization in Modern Intelligence, Archives / Vol. 2 No. 2 / Research Article, Published, 2020-11-28 DOI: https://doi.org/10.32996/jcsts.2020.2.2.4
11. A. Srivastava, "Securing PHI in cloud-based healthcare systems: Challenges, frameworks, and future directions," J. Comput. Anal. Appl., vol. 33, no. 2, Aug. 2024. [Online]. Available: https://www.eudoxuspress.com/index.php/pub/article/view/4349/3190
12. A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: https://ijcnis.org/index.php/ijcnis/article/view/8682
13. A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4347/3189
14. A. Srivastava, "A unified framework for secure cloud data management: Integrating governance, metadata intelligence, and privacy controls," J. Inf. Syst. Eng. Manage., vol. 9, no. 2, Apr. 2024. [Online]. Available: https://www.jisem-journal.com/download/45_A_Unified_Framework.pdf
15. A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4603
16. Abdul Aziz. “Intelligent Health Analytics: Mastering AI in Healthcare Data”. Acta Scientiae, vol. 25, no. 4, Aug. 2024, pp. 544-55, https://www.periodicos.ulbra.org/index.php/acta/article/view/687.

https://doi.org/10.5281/zenodo.21699886
17. Event-Aware Multi-Layer Storage Risk Forecasting for Oracle Database Estates Using HAPF, Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 10(4), 2024 ,10.22399/ijcesen.5183
18. Backup Integrity and Recovery Readiness Assessment for High-Availability Databases Raghu Gollapudi, Computer Fraud & Security, Vol. 2023, Issue 8, pp. 35-48, 10.52710/cfs.1031
19. Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 8(3), 2022 ,10.22399/ijcesen.5382
20. An Automated Risk Scoring Framework for SQL Execution Plan Analysis and Performance Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep),Publication year 2021, Volume 2021, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/970
21. Manoj Kumar Kagitha, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/969
22. Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS),Publication year 2023, Volume 2023, Issue 12,https://computerfraudsecurity.com/index.php/journal/article/view/971
23. Manoj Kumar Kagitha, LOAD BALANCING TECHNIQUES IN DISTRIBUTED CLOUD ENVIRONMENTS, (IJESMR - 2017), D.O.I:10.53555/jcr.v6:i1.13380, https://jcreview.com/archives/volume-6/issue-1/13380
24. Manoj Kumar Kagitha, AI-DRIVEN OBSERVABILITY FOR HYPERSCALE COLOCATION DATA CENTERS: A CASE STUDY OF LOG ANALYTICS AND ANOMALY DETECTION PIPELINES, Archives / Vol. 27 No. 7 (2019): JOCAAA,https://www.eudoxuspress.com/index.php/pub/article/view/5053
25. Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, https://www.eudoxuspress.com/index.php/pub/article/view/5054
26. Bandari, V. V., & Lekkala, H. B. (2024). AI-based operator behavior monitoring and cost optimization using digital traceability in manufacturing systems. Power System Protection and Control, 52(1), 38–45. https://pspac.info/index.php/dlbh/article/view/342
27. Venumadhav Vavilala, Shankar Balla (2024, May). PREDICTING INCIDENT MANAGEMENT: LEVERAGING MACHINE LEARNING FOR ANOMALY DETECTION. Power System Protection and Control Scopus Q1 Journal. PSPC. https://pspac.info/index.php/dlbh/article/view/270
28. Manikantha Varaprasad Inakollu, (2024, May), FINOPS-Driven Cloud optimization models for enterprise applications, Vol. 52 No. 2 (2024): April-June 2024, 99-110, Power System Protection and Control, ISSN-1674-3415, URL: https://pspac.info/index.php/dlbh/article/view/283,, DOI: https://doi.org/10.46121/pspc.52.2.10
29. Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. https://doi.org/10.5281/zenodo.13325989
30. Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2024-2027. https://dx.doi.org/10.21275/SR24724151733, https://www.ijsr.net/getabstract.php?paperid=SR24724151733
31. Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. https://doi.org/10.5281/zenodo.13348695
32. Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. https://doi.org/10.5281/zenodo.13325845
33. Jayanth Para, 6G Internet Technology Cyber Threat Notification & Alert System, Vol. 52 No. 4 (2024): October-December 2024, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/170 , DOI: https://doi.org/10.46121/pspc.52.4.9

34. Jayanth Para, AI Based Cloud Computation Observational Method & Process, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/171 , DOI: https://doi.org/10.46121/pspc.51.4.3
35. Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294
36. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364
37. Kaleshwar Aryasomayajula, Micro services: "A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments., Vol. 51 No. 2 (2023): April-June 2023 , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/374
38. Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/359
39. Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/362
40. Vikas Suresh Bagora, SECURE EMBEDDED NETWORKING ARCHITECTURES USING TRUSTED EXECUTION ENVIRONMENTS IN BROADBAND GATEWAYS, Vol. 52 No. 2 (2024): April-June 2024, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/363
41. Santhosh Chitraju Gopal Varma, The Evolution of Cloud-Native Architectures: Exploring the Synergy between Kubernetes and Microservices, DOI: https://doi.org/10.63282/3050-9246.IJETCSIT-V1I4P104, Vol. 1 No. 4 (2020), https://ijetcsit.org/index.php/ijetcsit/article/view/108
42. Santhosh Chitraju Gopal Varma, Artificial Intelligence in Cloud Computing: Building Intelligent, Distributed, and Fault-Tolerant Systems, Vol. 3 No. 1 (2022), DOI: https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P105, https://ijaibdcms.org/index.php/ijaibdcms/article/view/78
43. Sunil Karmakar, AI driven Identity & Access Management with Oracle Cloud security, Vol. 30 No. 2 (2022): JOCAAA, https://eudoxuspress.com/index.php/pub/article/view/4118
44. Rakshitha Kiran P., Samitha Khaiyum, Aparna R. Palandye, Apoorva S. D., Arjun R., Akshaya S. A., Annapoorna S., Leveraging LLaMA3 and LangChain for Rapid AI Application Development, Vol. 20 No. 10s (2024), https://doi.org/10.52783/jes.5539, https://journal.esrgroups.org/jes/article/view/5539.
45. Kancharla Vijayalakshmi, Bethapudi Samuel Anand Andrews, Estimation of Cyproheptadine Hydrochloride and Tricholine Citrate Simultaneously in Syrup Dose by Applying Stability Indicating HPLC Methodology, December 31, 2020, Vol. 11, Issue 4, Pages 1813–1819, DOI: 10.26452/ijrps.v11iSPL4.4383, https://ijrps.com/v11si4/3683
46. Vijayalakshmi Kancharla ,Kumar Abbaraju V D N, Andrews B S A, Nagaraju Dasari, Nagu Korupolu, Melphan as Anticancer and New Insights of Drug as Parenteral Dosage form against SARS-CoV-2 by Using Development and Validation , November 18, 2020, Vol. 11, Issue 1, Pages 1437–1446
DOI: 10.26452/ijrps.v11iSPL1.3686, https://ijrps.com/v11si1/2911
47. Ankit Pathak, Leveraging AI for Better Data Quality and Insights, Volume 7 Issue 3 (2025), https://doi.org/10.32996/jcsts.2025.7.2.19, https://al-kindipublisher.com/index.php/jcsts/article/view/9345
48. Uma Rajeswari Batchu, Bharati Reddi, Joshna Rani Surapaneni, Aruna Jangam, Sunil Misra, Anthony Addlagatta & Prakasham Reddy Shetty, Inhibition of Xanthine Oxidase by 1-O-Methyl Chrysophanol, A Hydroxyanthraquinone Isolated from Amycolatopsis Thermoflava ICTA 103, Published: 13 December 2025,Volume 66, pages 1049–1068 (2026)
https://link.springer.com/article/10.1007/s12088-025-01534-4