

Applied problems of information systems operation

UDC 004.9:004.72

doi: <https://doi.org/10.20998/2522-9052.2026.2.12>Eduard Malokhvii¹, Heorhii Kuchuk¹, Andrii Korobko², Nataliia Zhovnir³, Serhii Oryshchenko⁴¹ National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine² Kharkiv branch of Leonid Pohorilyy Ukrainian Research Institute for Forecasting and Testing of Agricultural Machinery and Technologies, Kharkiv, Ukraine³ National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine⁴ National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine

MULTI-OBJECTIVE FRAMEWORK FOR END-DEVICE PROCESSING AND OFFLOADING IN INDUSTRIAL IIOT

Abstract. The paper develops an integrated theoretical and mathematical framework for information processing on resource-limited Industrial Internet of Things (IIoT) end devices operating within cloud–fog–edge architectures. **The study is motivated** by heterogeneous, nonstationary event streams whose direct transmission to upper tiers is often infeasible due to bandwidth scarcity, strict latency targets, and the energy and computational limitations of end devices. Consequently, the end device must execute sensing-driven preprocessing, manage finite-buffer queues, and regulate outgoing traffic while preserving the informativeness required for monitoring, control, and analytics. The proposed formalization treats the end device as an active decision node that shapes system dynamics by controlling local transformations and offloading decisions under time-varying resource conditions. A class- and priority-aware stream model captures heterogeneity in criticality and service requirements, while finite-buffer queueing dynamics represent delay and loss under bursty arrivals and constrained service capacity. The device state is described by a resource vector reflecting available CPU capacity, memory and buffer occupancy, channel quality and transmission rate, and energy-related limitations, enabling state-dependent admissibility conditions for local computation and communication. An operator-level processing chain systematizes the end-device reduction pipeline, including preprocessing, informativeness assessment, adaptive filtering, temporal and semantic aggregation, controlled compression, and compact feature formation. The chain produces structured, semantically annotated packets supporting lightweight local decision-making and selective offloading to fog or cloud tiers. A multi-criteria efficiency structure is specified to jointly account for latency, packet loss, energy expenditure, communication load, and informativeness preservation, thereby enabling Pareto-oriented synthesis of admissible adaptive policies. **The research objective** is to establish unified decision variables, constraints, and stability and feasibility conditions coupling queue behavior with resource limitations, providing an analytically traceable basis for subsequent method construction, parameter tuning, and scenario-driven validation in realistic industrial environments. **Unlike purely** empirical benchmarking, the contribution is intentionally analytical: it consolidates fragmented models of local reduction and offloading, and exposes explicit operator definitions for reproducible analysis.

Keywords: Industrial Internet of Things; Cloud-Fog-Edge Computing; Adaptive Filtering; Event Aggregation; Controlled Compression; Queueing Model; Resource-State Modeling; Computation Offloading; Multi-Criteria Optimization.

Introduction

The rapid development of the Industrial Internet of Things (IIoT) has substantially transformed contemporary approaches to data acquisition, monitoring, and control in cyber-physical production environments [1, 2]. In contrast to conventional industrial automation systems, modern IIoT infrastructures are characterized by dense sensor deployment, continuous telemetry generation, event-driven signaling, and heterogeneous communication channels, which jointly produce high-volume and structurally diverse data streams [3, 4]. Under these conditions, the efficiency of the overall system increasingly depends not only on cloud-side analytics or network-level orchestration, but also on the capability of end devices to perform meaningful local processing before data are propagated to upper computational layers [5, 6].

The integration of cloud, fog, and edge computing has become a dominant architectural paradigm for IIoT because it enables the distribution of computational functions according to latency requirements, resource

availability, and application criticality [7, 8]. Cloud platforms provide scalable storage and high-performance analytics, while fog nodes offer intermediate processing capacity closer to production sites, and edge/end devices ensure low-latency interaction with physical processes [9]. However, the practical effectiveness of such a layered architecture is constrained by the weakest and most resource-limited components, namely end devices operating at the technological boundary. These devices must process incoming events under strict constraints on processor performance, memory capacity, communication bandwidth, and energy budget, while still maintaining the timeliness and reliability of critical industrial functions [10, 11].

Despite a significant body of literature devoted to resource management in cloud-fog-edge systems, many existing studies address optimization problems primarily at the fog or cloud levels, where computational resources are relatively abundant and end devices are often modeled as simplified traffic sources with fixed statistical properties [12, 13]. Such abstractions are suitable for global scheduling and network-wide load

balancing, but they are insufficient for the synthesis of end-device information processing strategies, where local filtering, aggregation, prioritization, and offloading decisions directly affect queue dynamics, communication load, and the availability of downstream services [13, 14]. In industrial scenarios, local decisions made at the edge can alter the intensity, composition, and temporal structure of traffic reaching fog and cloud nodes, thereby influencing not only local responsiveness but also the global operational stability of the entire distributed architecture [15].

A key methodological difficulty in this context is the need to represent end-device input as a heterogeneous event stream rather than as a homogeneous sequence of measurements. Industrial data flows typically combine periodic telemetry, asynchronous alarm messages, control commands, diagnostic markers, protocol service packets, and software-level events, each with distinct arrival patterns, priorities, and informational value [16, 17]. Their statistical properties are often nonstationary and depend on equipment states, operating modes, environmental conditions, and abnormal situations, which limits the applicability of oversimplified stationary traffic models [18]. Therefore, a rigorous formalization of end-device information processing requires a stochastic event-based model capable of capturing class-dependent arrivals, temporal variability, queue occupancy, and the interaction between local computation and communication constraints [19].

Another important challenge is that end-device processing in IIoT cannot be formulated as a single-objective optimization problem. Any strategy that aggressively reduces traffic and computation cost through filtering and aggregation may simultaneously decrease the informational content of retained data, thereby degrading diagnostics and control quality [20, 21]. Conversely, a strategy aimed at maximal data preservation tends to increase queue lengths, latency, packet loss probability, and energy consumption, especially in resource-constrained microcontroller-based platforms [22]. As a result, the synthesis of information processing mechanisms at the end-device level must be treated as a multi-criteria problem, where latency, loss, energy expenditure, communication load, and informativeness are considered jointly and balanced under explicit resource constraints [23, 24].

From this perspective, the development of a mathematically grounded framework for end-device information processing requires the integration of several methodological components. First, the physical process observed by an end device should be linked to a discrete event representation that includes measured attributes, timestamps, event classes, and priority levels [25]. Second, local processing must be represented through queueing-theoretic abstractions that account for finite buffers, stochastic service times, and overload behavior for different classes of events [26]. Third, the resource state of the end device should be modeled as a dynamic vector reflecting processor utilization, queue state, communication channel conditions, and energy availability. Finally, system performance must be described through a generalized set of indicators that

jointly characterize service quality, resource consumption, and information preservation efficiency. Such an integrated formalization creates the theoretical basis for constructing adaptive processing rules for filtering, preprocessing, aggregation, and computational offloading in layered IIoT environments [27, 28].

In addition to architectural and queueing considerations, the preprocessing stage performed directly on end devices plays a fundamental role in determining the structure and usefulness of data delivered to higher layers. In industrial applications, raw sensor values are subject to measurement noise, quantization errors, communication disturbances, missing samples, and calibration drift, which necessitates local operations such as outlier detection, noise suppression, normalization, time alignment, and missing-data reconstruction [29]. These operations are not merely auxiliary; they define the initial transformation of raw observations into a compact and semantically meaningful event stream that can be processed under strict computational constraints [30]. Consequently, preprocessing and adaptive filtering should be analyzed as integral elements of the end-device information processing problem rather than as independent engineering routines [31].

The purpose of this paper is to formulate a coherent theoretical and mathematical foundation for information processing on IIoT end devices within a cloud-fog-edge architecture, with emphasis on the formal representation of event flows, local queue-based processing, resource-aware decision variables, and multi-criteria efficiency indicators [32, 33]. Unlike experimental studies focused on empirical benchmarking, the present work is intentionally centered on conceptual and analytical synthesis: it systematizes the problem domain, introduces a unified modeling perspective, and defines the structure of the optimization problem that underlies adaptive end-device processing in industrial environments [34]. Such a formulation is necessary for subsequent development, analysis, and validation of specialized methods capable of improving responsiveness, reducing communication load, and preserving the informational value of industrial data under realistic edge constraints [35].

Internet of Things (IoT) is emerging as a pivotal advancement within the realm of Artificial Intelligence (AI), signifying a stride towards a transformative paradigm in contemporary society [36]. In this context, the term “things” pertains to tangible devices, thereby constituting a network comprised of sensors and processors facilitating data communication among these devices [37]. Within this network, two categories of nodes exist: physical nodes and virtual nodes. Physical nodes encompass sensors, actuators, transit nodes, and other wearable or embedded devices, while virtual nodes comprise virtual machines or networks utilized within wireless networks.

1. Related work

The evolution of IoT and IIoT computing infrastructures has led to a gradual transition from purely cloud-centric architectures toward distributed cloud-fog-edge paradigms, where computational functions are

placed across multiple layers according to latency, bandwidth, and reliability requirements [38]. In early cloud-oriented approaches, end devices were primarily considered data sources, while the main analytical and decision-making functions were concentrated in remote data centers. This model remains effective for long-term storage, large-scale analytics, and computationally intensive optimization; however, in industrial environments it often introduces unacceptable communication delays, excessive upstream traffic, and limited resilience under unstable network connectivity [39]. The reviewed literature consistently emphasizes that cloud-only processing is insufficient for time-sensitive industrial applications, especially when physical processes require deterministic response and continuous state awareness [40].

To address these limitations, fog computing has been introduced as an intermediate layer that extends cloud capabilities toward the network edge by providing geographically closer processing, buffering, and service orchestration [41]. Fog nodes are typically used to aggregate data from multiple devices, perform near-real-time preprocessing, and reduce the amount of information sent to the cloud. In IIoT contexts, this paradigm is widely applied to predictive maintenance, process monitoring, and distributed control support, where partial local intelligence is needed but full cloud dependence is undesirable [42]. Nevertheless, many fog-oriented studies still formulate the end device as a simplified traffic generator or as a terminal node with fixed behavior, which limits the analytical expressiveness of such models when the objective is to design adaptive processing directly on the end device [43]. In other words, fog-based architectures improve the placement of computation, but they do not always provide a sufficiently detailed formal model of event handling, queueing, and resource-aware decision making at the lowest level of the hierarchy [44].

Edge computing research further strengthens the role of local processing and pushes decision logic closer to sensors and actuators [45]. The edge paradigm is particularly relevant for IIoT because it supports low-latency operation, local autonomy, and reduced communication overhead, all of which are critical in production systems with high event rates and strict timing constraints [46]. Existing edge-focused studies propose a wide range of mechanisms, including local anomaly detection, threshold-based event filtering, stream preprocessing, and task partitioning between edge and upper layers [47]. However, these mechanisms are often developed as domain-specific solutions for particular sensors, protocols, or application scenarios. As a result, many approaches remain fragmented: one group of works focuses on event filtering, another on resource scheduling, another on communication-efficient aggregation, and another on offloading policies, while a unified mathematical representation that integrates these stages at the level of a resource-constrained IIoT end device is rarely presented explicitly [48].

A separate line of research addresses data reduction techniques, including filtering, aggregation, compression and feature extraction, which are especially important in

industrial telemetry systems with large volumes of redundant or weakly informative observations [49]. Threshold filtering, event suppression, and dead-band transmission are widely used to decrease communication load, but they can also cause loss of diagnostically relevant information when static rules are applied under dynamic operating conditions [50]. Temporal and event-based aggregation methods significantly reduce the number of transmitted records and are effective for trend-level monitoring, yet they may distort short transients, rare faults, and sequence-dependent patterns if aggregation windows or grouping logic are not adapted to the current process state [51]. Compression methods improve channel utilization but introduce an additional trade-off between bitrate reduction and data distortion, which becomes critical when compressed data are later used for diagnostics or control [52]. Feature extraction techniques mitigate part of this problem by transforming raw or aggregated signals into compact descriptors, but their computational overhead and informational adequacy depend strongly on the available local resources and the semantics of the monitored process [53].

In parallel, task offloading and computation placement remain major topics in edge/fog/cloud research, especially in relation to delay minimization and energy-aware scheduling [54, 55]. Most offloading models optimize execution location based on communication latency, computational cost, or energy expenditure, and many of them are formulated for mobile edge computing or generalized IoT scenarios rather than industrial event-driven systems [56]. While these models provide valuable optimization principles, they frequently assume that the data to be transmitted are already formed and do not explicitly model the upstream transformation pipeline on the end device, including preprocessing quality, informative filtering, and adaptive aggregation [57]. Consequently, the interaction between local data reduction and offloading decisions is often underrepresented. For IIoT applications, this omission is substantial, because the decision to offload depends not only on network and CPU conditions, but also on the informational value of the event or packet being considered and on the consequences of local simplification for process observability [58].

Another limitation observed in the literature is the tendency to employ homogeneous or weakly differentiated traffic abstractions, which are insufficient for industrial event streams composed of telemetry, alarms, service packets, and control-related messages with different priorities and temporal behavior [59]. In practical IIoT systems, event classes differ in arrival intensity, urgency, and semantic significance, and these differences directly affect queue occupancy, service discipline, and packet loss under overload conditions [60]. Models that do not distinguish event classes can be useful for coarse-grained capacity estimation, but they are inadequate for constructing adaptive local policies that must preserve high-priority and high-informativeness data while restricting low-value traffic under resource pressure. The need for class-aware and state-aware modeling is therefore not only a theoretical preference but a functional requirement for robust end-device information processing [61, 62]. This requirement

is also consistent with formalization of heterogeneous event streams, local queue dynamics, and resource-state vectors in cloud-fog-edge settings, which provides a stronger basis for end-device decision synthesis than generic traffic models [63].

A further methodological issue concerns the treatment of performance criteria. Many studies optimize one or two indicators, most commonly latency and energy, whereas the preservation of informational value is either omitted or treated implicitly [64]. This simplification is understandable in network-centric and scheduling-centric formulations, but for IIoT end-device processing it creates a significant gap because local filtering and aggregation inevitably modify the informational structure of the transmitted data. An efficient communication policy that minimizes traffic and local computation can still be unacceptable if it suppresses weak precursors of faults or distorts process dynamics needed for diagnosis [65]. Conversely, full data preservation may exceed the device's computational or energy capacity and lead to unstable queues and delayed reactions. Therefore, the research problem should be framed as genuinely multi-criteria, with explicit inclusion of latency, packet loss, energy consumption, communication load, and informativeness in a unified analytical model [66]. The formal definitions of such criteria and their integration into a single efficiency vector can be systematically generalized for article-level presentation.

Based on the above analysis, the principal research gap can be formulated as follows. Existing studies provide important but largely disjoint contributions to cloud-fog-edge architecture design, local data reduction, and offloading optimization; however, they rarely offer a coherent mathematical framework that simultaneously represents heterogeneous event-based input on IIoT end devices, local queue-constrained processing, dynamic resource state of the device, adaptive transformations of data such as filtering, aggregation, compression, and feature formation, and a multi-criteria decision basis for selecting local processing versus transmission to upper layers [67]. In particular, the coupling between event informativeness and resource-aware decision making remains insufficiently formalized in most available models [68]. This gap limits the reproducibility, comparability, and systematic development of adaptive end-device processing mechanisms for industrial applications.

The paper addresses this gap by developing a unified theoretical and mathematical formalization of information processing on IIoT end devices within a cloud-fog-edge architecture. Instead of proposing an empirical benchmark or a narrowly specialized implementation, the paper structures the problem domain into a sequence of interconnected models: an event and stream representation, a queue and service model, a resource-state model, a set of adaptive data processing stages, and a multi-criteria decision framework for local processing and offloading [69, 70]. Such a formulation is intended to serve as a methodological foundation for subsequent synthesis and validation of adaptive strategies in realistic industrial scenarios, while

preserving analytical clarity and explicit traceability of assumptions.

2. Problem formulation

The object of research in this paper is the process of information processing on Industrial Internet of Things end devices operating within a hierarchical cloud-fog-edge architecture. In such a system, end devices form the primary interface to physical processes and continuously interact with upper computational layers through communication channels of limited and potentially variable capacity (Fig. 1). From the standpoint of system organization, the end device is not treated as a passive data source, but as an active processing node whose local decisions influence queue dynamics, communication load, and the temporal structure of data delivered to fog and cloud layers. This interpretation is consistent with the formal architectural decomposition of edge, fog, and cloud node sets and with the event-oriented representation of end-device operation adopted in the underlying framework.

The subject of research is the set of mathematical models and decision-making principles that govern adaptive selection of end-device processing modes under resource constraints. In the scope of this paper, these modes include local preprocessing, adaptive filtering, aggregation, compression, feature-oriented transformation, and transmission of data to upper layers for further processing. The emphasis is placed on the formal coupling between input event characteristics and the current resource state of the device, rather than on a specific implementation platform or protocol stack. Accordingly, the paper focuses on a generalized analytical formulation that can later be instantiated for concrete IIoT domains, hardware classes, and communication technologies.

The system boundary considered in this study includes the end device as the principal decision point, the local communication interface as a constrained transfer medium, and upper-level fog/cloud resources as available destinations for deferred or extended processing.

At the same time, low-level protocol details, vendor-specific hardware features, and application-specific semantic interpretations of events are intentionally abstracted and represented through generalized model parameters. This abstraction is necessary to preserve analytical generality while retaining the essential dynamics of event arrivals, local processing, and adaptive transmission behavior.

2.1 Input event model and operating constraints.

The input to the considered system is a heterogeneous event stream generated by industrial sensors, actuators, controllers, and service software components. In contrast to homogeneous telemetry models, the stream is assumed to include both periodic and asynchronous events with different semantic roles, priorities, and temporal distributions. This assumption reflects the operational reality of IIoT systems, where regular measurements coexist with alarms, service packets, protocol-level events, and diagnostic markers, and where event intensity may vary depending on process state and disturbance conditions.

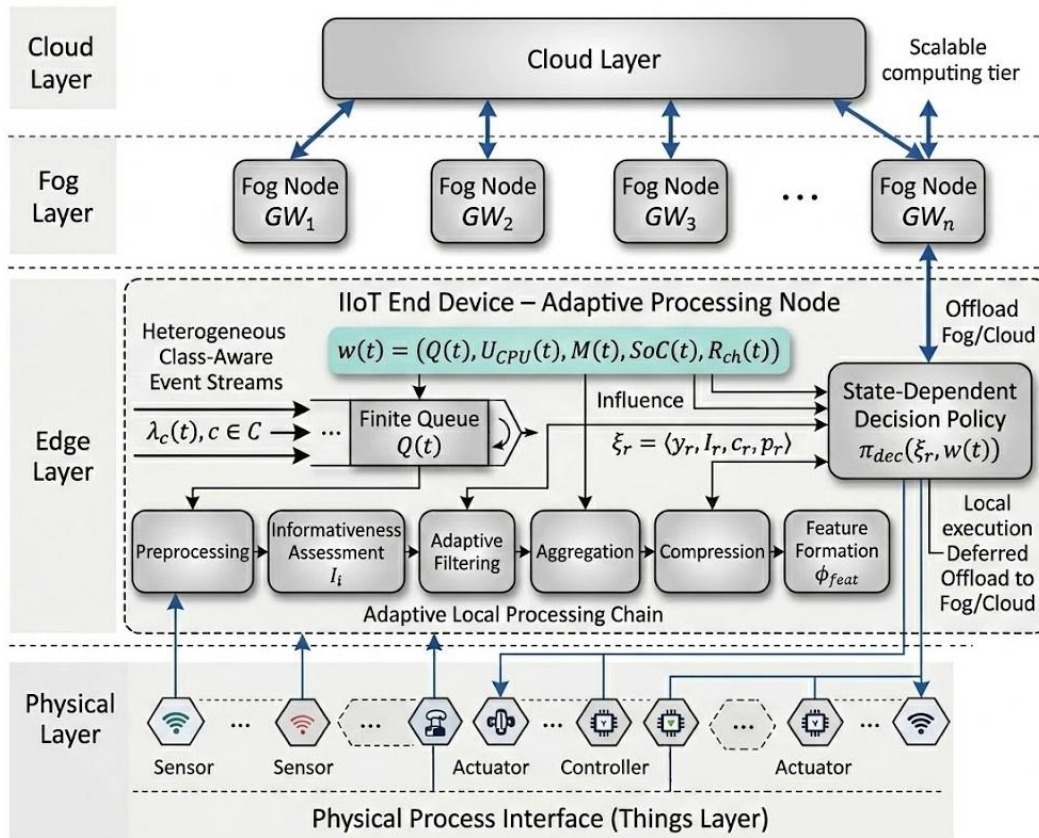


Fig. 1. Cloud-fog-edge IIoT architecture featuring the end device as an active decision node

The considered framework formalizes such data as class-dependent event flows, thereby establishing a basis for an event-driven problem formulation rather than for a simplified scalar-sampling model.

The considered end device operates under explicit resource constraints that significantly affect feasible processing strategies. These constraints include limited computational capacity, finite memory and buffer space, limited or unstable communication throughput, and restricted energy availability. In practical terms, this means that the device cannot always execute full local processing for every incoming event and cannot transmit all raw data to upper layers without risking queue growth, delay escalation, or energy depletion. Therefore, any viable processing policy must dynamically reconcile the incoming event load with the current operational capacity of the device. In the mathematical treatment that follows, these constraints will be captured through a dynamic resource-state representation and queue-aware processing assumptions.

In addition to resource limitations, the problem is constrained by operational requirements typical for industrial systems. These include bounded latency for time-sensitive events, admissible loss limits (which may be class-dependent), and continuity requirements for baseline telemetry delivery to ensure monitoring traceability and diagnostic observability. The latter condition is especially important because aggressive filtering or aggregation may reduce communication load but can violate minimum reporting obligations if applied without safeguards. For this reason, the problem statement explicitly assumes the existence of regulatory

or operational constraints that impose a lower bound on retained and transmitted information, even under overload conditions.

2.2 Performance criteria and tradeoffs. The efficiency of end-device information processing in IIoT cannot be adequately described by a single indicator. The problem is inherently multi-criteria because local decisions simultaneously affect processing delay, queue occupancy, packet or event loss, energy consumption, communication traffic, and the informativeness of the retained data. In industrial environments, these indicators are not independent: they interact through both computational and communication pathways, and the improvement of one criterion often leads to deterioration of another. The integrated efficiency vector formalization provides a suitable foundation for representing this interdependence in a systematic way.

The central conflict of the problem can be summarized as a trade-off between resource efficiency and information preservation. Stronger local filtering, larger aggregation windows, or more aggressive compression settings tend to reduce traffic volume and energy use, and they may also stabilize queue behavior under heavy load. However, these same actions can suppress weak but diagnostically important events, distort temporal dynamics, or remove context required by upper-level analytics. Conversely, preserving the full event stream or applying minimal reduction improves observability but increases the burden on the local processor, communication channel, and energy subsystem, thereby elevating latency and overload risk. The problem is further complicated by the fact that the

informational value of events is itself nonuniform and may depend on event class, process condition, and short-term event history, which makes static processing rules insufficient for robust operation.

Under such conditions, the decision mechanism must be adaptive and state-aware. This means that processing actions should not be fixed a priori, but selected dynamically on the basis of current event characteristics and the instantaneous resource state of the device. In particular, the system should support variation of filtering intensity, aggregation mode, compression level, and transmission behavior in response to changes in queue occupancy, processor load, energy reserve, and communication conditions. End-device processing is therefore treated as an adaptive control problem over a structured set of data transformation and offloading actions, rather than as a fixed pipeline with constant parameters.

2.3 Multi-objective optimization problem. Let $\theta(t)$ denote a vector of adaptive control variables (or policy parameters) that determines the current end-device processing configuration. In a generalized form, this vector may include parameters of local filtering, aggregation, compression, feature formation, and transmission/offloading behavior, i.e.:

$$\theta(t) = (\theta_{\text{filt}}(t), \theta_{\text{agg}}(t), \theta_{\text{cmp}}(t), \theta_{\text{feat}}(t), \theta_{\text{off}}(t)).$$

The problem is to determine an admissible adaptive policy π that maps input events and the current device state to feasible values of $\theta(t)$, while satisfying operational and resource constraints. The optimization objective is multi-criteria and can be expressed through a vector of performance functionals that jointly represent latency, loss, energy expenditure, communication load, and informativeness. In compact form, the problem can be written as a vector optimization task over the admissible policy set Π :

$$\min_{\pi \in \Pi} \mathbf{F}(\pi) = \min_{\pi \in \Pi} (f_T(\pi), f_L(\pi), f_E(\pi), f_R(\pi), -f_I(\pi)),$$

where f_T is the delay-related criterion, f_L characterizes losses, f_E reflects energy consumption, f_R denotes communication load, and f_I represents informativeness (introduced with a negative sign to preserve minimization form). Depending on the target application, this formulation may be treated either as a Pareto optimization problem or as a constrained scalarized problem with prioritized criteria and admissible thresholds.

The admissible decision space is restricted by a set of constraints reflecting end-device computational capacity, memory and buffer limits, communication channel conditions, energy budget, and minimal telemetry preservation requirements. In addition, service-level restrictions may impose upper bounds on acceptable delay for selected event classes. Thus, the research problem is not merely to reduce traffic or local load, but to construct a formally consistent and resource-aware decision framework that preserves the functional adequacy of industrial monitoring and control under variable operating

conditions. To enable analytical development of this framework, the next section introduces the mathematical model of event streams, queue-based local processing, device resource state, and efficiency criteria that underlies the proposed formalization.

3. System model for IIoT end-device information processing

The proliferation of IoT technologies has ushered in a transformative era, permeating diverse sectors with unprecedented connectivity and data generation. In tandem with this IoT revolution, the integration of cloud, fog, and edge computing has emerged as a pivotal paradigm, reshaping the landscape of computational architecture [11]. This integration represents a strategic response to the escalating demands for real-time data processing, low-latency communication, and scalable infrastructure in the IoT ecosystem.

3.1 Event stream model. Information processing at an IIoT end device is formalized as the handling of a heterogeneous event stream generated by sensors, actuators, controllers, and service subsystems operating in a shared industrial environment. Such a representation is required because the incoming traffic is not homogeneous telemetry only; it also includes alarms, diagnostic messages, control-related updates, and auxiliary protocol events that differ in temporal behavior, criticality, and processing requirements. A unified event abstraction is therefore introduced in order to preserve semantic heterogeneity while maintaining a mathematically tractable model for subsequent optimization and control.

Each incoming event is represented by the tuple:

$$x_i = \langle v_i, s_i, c_i, p_i \rangle,$$

where v_i denotes the measured value (or feature vector), s_i is the timestamp, c_i is the class label, and p_i is the priority (or criticality indicator). This structure supports simultaneous modeling of signal content, temporal ordering, semantic type, and service importance, which is essential for industrial scenarios where not all events should be processed, compressed, or transmitted under identical rules. The adoption of a class- and priority-aware event model is consistent with a formulation in which adaptive filtering, aggregation, compression, and offloading are coupled and jointly influence queue dynamics, delays, energy use, and retained informativeness.

Let $\mathcal{C} = \{1, \dots, C\}$ denote the set of event classes. The input stream is modeled as a superposition of class-specific substreams with time-varying intensities $\lambda_c(t)$, $c \in \mathcal{C}$, and aggregate intensity:

$$\lambda(t) = \sum_{c \in \mathcal{C}} \lambda_c(t).$$

The explicit use of class-dependent intensities is justified by the operating reality of IIoT systems: the traffic profile changes with process regimes, maintenance operations, faults, communication disturbances, and control actions. Consequently, the composition of the stream is often more important than

the total rate alone. A purely aggregate model would obscure the distinction between critical and background traffic, which is unacceptable in industrial processing where latency and loss constraints are class-dependent.

To account for selective local processing, an admissibility operator $\alpha(x_i, w(t), \theta) \in 0,1$ is introduced, where $w(t)$ is the instantaneous resource state and θ is the control configuration (filtering, aggregation, compression, and task-formation parameters). The effective locally admitted stream is then defined as:

$$\mathcal{X}_{loc} = \{x_i : \alpha(x_i, w(t_i), \theta) = 1\},$$

and its intensity is denoted by $\lambda_{eff}(t)$. This distinction between the raw incoming stream and the admitted processing stream is central for the formal model, because most adaptive mechanisms act precisely through transformations of $\lambda_{eff}(t)$, event granularity, and the information density of the retained data rather than through direct modification of the physical sensing process itself.

A horizon-based representation is also convenient for deriving performance functionals. For an observation interval $[t_0, t_0 + H]$, the event sequence is represented as:

$$\mathcal{X}_H = \{x_i | s_i \in [t_0, t_0 + H]\}.$$

This enables the later introduction of horizon-level criteria (delay, loss, energy, bitrate, informativeness) as functionals over realized trajectories instead of static point values. Such a formulation aligns with the multi-criteria treatment of end-device operation, where strategy quality is assessed through integrated indicators rather than isolated instantaneous metrics.

3.2 Queueing and local processing model. Local processing at the end device is represented by a queue-based service model with finite buffering and resource-constrained service capacity. This abstraction is necessary because even when filtering and aggregation are applied, event handling remains subject to contention for processor time, memory, and communication interface access.

Queue state and deadlines are explicitly involved in the interaction of adaptive filtering and offloading decisions, which motivates a queue-centric mathematical representation of local execution.

Let $Q(t)$ denote the queue occupancy at time t , measured either in events or in normalized service units. A generic discrete-time evolution over step Δt may be written as:

$$Q(t + \Delta t) = \min\{Q_{\max}, \max\{0, Q(t) + A(t, \Delta t) - S(t, \Delta t)\}\},$$

where $A(t, \Delta t)$ is the number (or workload volume) of admitted arrivals during $[t, t + \Delta t)$, $S(t, \Delta t)$ is the completed service volume, and Q_{\max} is the finite queue capacity. This form does not impose a restrictive stochastic assumption (e.g., Poisson/exponential service) and is therefore suitable for an analytical framework that

must remain compatible with heterogeneous industrial workloads and implementation-specific schedulers.

An effective service intensity $\mu(t)$ is introduced to represent the local processing capability under the current configuration and resource state. The corresponding utilization indicator is defined as:

$$\rho(t) = \lambda_{eff}(t) / \mu(t).$$

Although $\rho(t)$ is an aggregated quantity, it provides an operationally meaningful criterion for stability and overload detection. Regimes with sustained $\rho(t) \geq 1$ imply queue accumulation, increased waiting times, and elevated probability of loss or deadline violation, whereas $\rho(t) < 1$ on admissible intervals is required for bounded queue behavior. This interpretation is especially important in a framework where adaptive filtering and compression may reduce $\lambda_{eff}(t)$, while feature extraction or local analytics may simultaneously decrease $\mu(t)$ by increasing per-event computational complexity.

To preserve class-sensitive processing behavior, the queue model is supplemented by class-indexed waiting-time and deadline constructs. Let D_i denote the sojourn time of event x_i , and let T_c^{\max} be the admissible delay for class c . Deadline violations are then characterized by the indicator:

$$\delta_i = \mathbf{1}\{D_i > T_{c_i}^{\max}\}.$$

This definition links queue dynamics to loss-related criteria in a manner that is more appropriate for IIoT than simple overflow-only loss modeling, because many industrial applications treat stale information as functionally equivalent to dropped information. The distinction between critical and background flows in the criteria vector further reinforces the need for class-aware accounting of delay and loss.

The queueing formulation also serves as the coupling point between local processing and offloading. Events that are not efficiently handled locally may be converted into offloading tasks after preprocessing, aggregation, and feature formation. Consequently, queue occupancy $Q(t)$ is not only a service-state descriptor but also a control signal that influences whether subsequent workload should remain on the end device or be transferred to fog/cloud layers. This interpretation aligns directly with the formulation in which offloading decisions are made with explicit consideration of queue state, deadlines, and resource constraints.

3.3 End-device resource state model. The local operating conditions of the IIoT end device are represented by a dynamic resource-state vector:

$$w(t) = (Q(t), U_{CPU}(t), M(t), SoC(t), R_{ch}(t)),$$

where $Q(t)$ is queue occupancy, $U_{CPU}(t)$ is processor utilization, $M(t)$ is memory usage (or memory pressure), $SoC(t)$ is the energy state, and $R_{ch}(t)$ is the available communication capacity (or an equivalent channel-

quality/throughput indicator). The explicit inclusion of memory extends the compact notation and is consistent with the criteria formulation in which the integrated resource load accounts for processor utilization, memory, and queue state.

A state-based formulation is indispensable because the admissibility of local processing and the feasibility of offloading cannot be determined from event properties alone. For instance, the same event class may be processed locally under low CPU load and high channel congestion, but routed outward when CPU saturation is high and channel conditions are favorable. Similarly, energy-constrained operation requires policies to react to $SoC(t)$, since computationally expensive local analytics and communication-heavy transmission exhibit different energy signatures. The source material explicitly treats offloading and upstream data preparation as a coordinated chain that is sensitive to queue state, processor load, energy, and channel characteristics, which is naturally captured by the resource vector $w(t)$.

The state vector also supports constraint formalization. Let \mathcal{W}_{adm} denote the admissible state set defined by operational and safety bounds:

$$\mathcal{W}_{adm} = \left\{ \begin{array}{l} w: Q \leq Q_{max}, U_{CPU} \leq U_{max}, \\ M \leq M_{max}, SoC \geq SoC_{min}, R_{ch} \geq R_{min}^{op} \end{array} \right\}.$$

This set is not merely a technical convenience; it encodes feasibility requirements under which the end device remains stable, responsive, and deployable in industrial environments.

Any processing policy that systematically drives the system outside \mathcal{W}_{adm} must be considered inadmissible even if it improves one isolated metric (e.g., informativeness) in the short term.

A practical implication of the state formulation is the need for synchronized measurement and estimation. Components such as $Q(t)$ and $U_{CPU}(t)$ are directly observable, while $R_{ch}(t)$ and effective transmission conditions may require short-term estimation or smoothing. The resulting state vector can therefore be interpreted as a hybrid measured-estimated quantity updated at policy decision epochs. This interpretation is consistent with treating the end device as an active control element that manages both information flow and resource consumption, rather than a passive data source.

3.4 Performance criteria vector. The performance of end-device information processing is described by a multi-criteria vector rather than a scalar objective. This is a foundational modeling choice: in industrial IIoT systems, latency, deadline adherence, energy expenditure, communication load, and data informativeness are inherently interdependent and often conflicting. An integrated criteria vector is formulated, and strategy synthesis is required to explicitly respect this coupled, joint structure. Let π denote the end-device processing strategy and define the integrated criteria vector as:

$$\mathbf{F}(\pi) = (F_T^{(crit)}, F_T^{(bg)}, F_{loss}^{(crit)}, F_{loss}^{(bg)}, F_E, F_I, F_U, F_R),$$

where the components correspond to delay for critical and background flows, loss/deadline-miss indicators for critical and background flows, energy expenditure, informativeness of retained/transmitted data, integrated resource utilization, and communication resource usage (e.g., average bitrate and peak channel load). This structure directly reflects the underlying formal decomposition of the criteria.

The delay components $F_T^{(crit)}$ and $F_T^{(bg)}$ can be defined as mean, percentile, or guaranteed latency metrics over the corresponding class subsets. The loss components $F_{loss}^{(crit)}$ and $F_{loss}^{(bg)}$ capture overflow losses and/or deadline violations. The energy criterion F_E is naturally formulated as energy per informative event, energy per horizon, or a weighted combination of compute and transmission consumption. The resource-load criterion F_U aggregates processor, memory, and queue pressure into an integrated operational burden measure. The channel-use criterion F_R characterizes both average and peak network usage, which is important in shared industrial communication environments. These interpretations are explicitly supported by the criteria description.

The informativeness criterion F_I plays a structurally distinct role. Unlike delay, loss, energy, and bitrate, which are naturally minimized, informativeness is a “useful” quantity that should be preserved or maximized. The paper notes this asymmetry and indicates that a unified minimization form may be obtained by normalization or by transforming utility-type criteria into equivalent penalty forms. This observation is important for the later optimization formulation because it permits a consistent Pareto analysis without distorting the semantic meaning of the original criteria.

A key methodological consequence of the vector formulation is that no single mechanism can be optimized independently without affecting the others. Adaptive filtering reduces the admitted stream and may improve F_T , F_E , and F_R , but excessive filtering degrades F_I . Aggregation and compression reduce bitrate and transmission costs yet may alter task fidelity and delay profiles. Offloading decisions may reduce local load F_U and local delay under CPU saturation, but can worsen latency or reliability under channel degradation.

3.5 State-policy-performance model. The complete mathematical model is obtained by integrating event-stream representation, queue dynamics, resource-state evolution, and the multi-criteria performance vector into a single controlled system (Fig. 2). In this formulation, the processing strategy is not treated as a fixed algorithmic routine but as a structured decision object that determines how incoming data are transformed and where computations are executed under changing resource conditions.

The end-device processing strategy is modelled as an ordered set of coordinated policies:

$$\pi = (\pi_{prep}, \pi_{filt}, \pi_{agg}, \pi_{cmp}, \pi_{off}),$$

where the individual components correspond to preprocessing/normalization, adaptive filtering, aggregation, compression, and offloading, respectively. This modular representation is essential for the mathematical formalization because it makes the internal control interfaces of the processing pipeline explicit and, consequently, enables systematic tuning of end-device behavior under varying resource and workload conditions.

The corresponding parameterization may be written as:

$$\theta = (\theta_{prep}, \theta_{filt}, \theta_{agg}, \theta_{cmp}, \theta_{task}, \theta_{off}),$$

where θ_{task} denotes the task-formation parameters used after aggregation/feature generation. The parameter vector is explicitly linked to changes in the integrated criteria and is treated as the primary object of synthesis.

State evolution is then expressed by a controlled transition relation:

$$w(t + \Delta t) = \Phi(w(t), \mathcal{X}(t, t + \Delta t), \pi, \theta),$$

where $\mathcal{X}(t, t + \Delta t)$ denotes the incoming events on the interval and $\Phi(\cdot)$ aggregates the effects of admission, queuing, local execution, data reduction, task formation, and transmission decisions.

A separate performance mapping:

$$\mathbf{F}(\pi) = \Psi(\{x_i\}, \{w(t)\}, \pi, \theta),$$

generates the integrated criteria over the observation horizon. The pair (Φ, Ψ) closes the model by explicitly connecting controllable transformations to measurable performance outcomes.

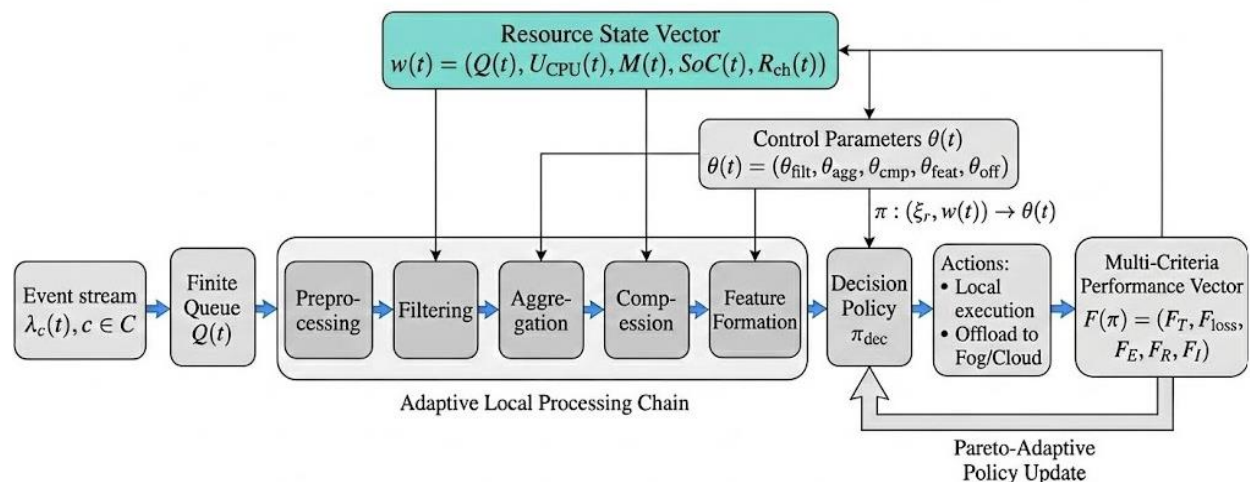


Fig. 2. State-policy-performance model for adaptive control

This integrated representation also clarifies why the problem is intrinsically multi-objective and state-dependent. The same control parameters θ can yield different performance vectors under different realizations of $\lambda(t)$, class composition, and resource-state trajectories $w(t)$. Consequently, a valid synthesis procedure cannot rely on static parameter tuning alone; it must account for workload variability, state transitions, and class-sensitive service requirements. This point is consistent with interpreting the end device as an adaptive controller of both data flows and resource usage in cloud-fog-edge environments.

Finally, the model provides a rigorous transition to the optimization problem. The admissible strategy set is defined by physical, operational, and deployment constraints (resource limits, deadline requirements, communication feasibility, minimal informativeness preservation), and the target of synthesis is a non-dominated compromise in the criteria space. The strategy search is framed in Pareto terms under stability, scalability, and realizability requirements, making the present state-policy-performance model an appropriate formal basis for the subsequent optimization section.

4. Adaptive data processing framework

Adaptive information handling on IIoT end devices requires a formal operator-level description of local data

transformations, since resource-aware behavior emerges not from a single mechanism but from the coordinated action of multiple processing stages that jointly reshape event intensity, representation granularity, and informative content. The local processing chain is therefore modeled as a sequence of parameterized transformations including preprocessing and normalization, informative-value assessment, filtering, aggregation, compression, and feature formation, where each stage has explicit inputs, outputs, and admissibility conditions [11] (Fig. 3).

This formulation preserves analytical consistency across heterogeneous industrial event streams and constrained device states, while providing a mathematically explicit basis for compact, semantically structured representations suitable for subsequent local analysis and controlled interaction with upper computational layers.

4.1 Preprocessing and data normalization. Local processing of IIoT data begins with a preprocessing layer that transforms heterogeneous raw observations into a structurally consistent representation suitable for subsequent filtering, aggregation, and decision-making. This stage is required because industrial event streams are affected by measurement noise, quantization artifacts, missing samples, timestamp jitter, protocol-level irregularities, and device-specific calibration offsets, all of which distort downstream estimates of

informativeness and may induce unstable behavior of adaptive rules if left untreated. A unified operator-based

formalization is therefore introduced in order to preserve analytical coherence across the full processing pipeline.

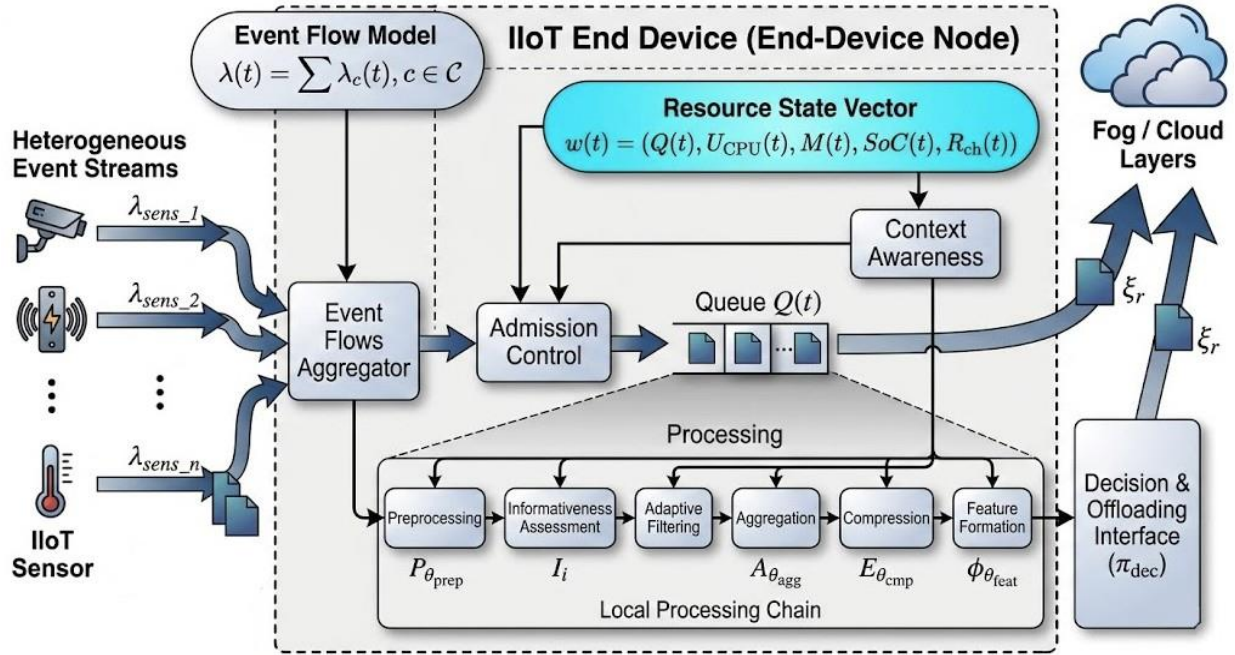


Fig. 3. Adaptive local processing chain for heterogeneous event streams on the IIoT end device

Let $x_i = \langle v_i, s_i, c_i, p_i \rangle$ denote the raw event defined in the previous section.

The preprocessed event is represented as:

$$\hat{x}_i = P_{\theta_{prep}}(x_i, H_i^{raw}),$$

where $P_{\theta_{prep}}$ is a parameterized preprocessing operator and H_i^{raw} is a short raw-history context used for local correction and synchronization.

The parameter block θ_{prep} may include denoising parameters, calibration coefficients, missing-value reconstruction settings, and synchronization tolerances. The resulting event \hat{x}_i preserves the semantic structure of x_i while providing a normalized value representation and consistent timing attributes.

The preprocessing operator is modeled as a composition of elementary transformations:

$$P_{\theta_{prep}} = P_{sync} \circ P_{norm} \circ P_{clean},$$

where $P_{\theta_{prep}}$ performs cleaning and basic correction, P_{norm} performs normalization/calibration, and P_{sync} aligns timestamps and event timing semantics.

The cleaning stage suppresses gross outliers and corrupted payloads using lightweight local rules compatible with constrained devices, e.g., bounded-range clipping, median-type smoothing on short windows, or class-specific validity masks. The normalization stage maps measured values to a common scale, compensates known sensor offsets, and may convert device-specific units into process-level units. The synchronization stage aligns events to a local time

base and resolves inconsistencies caused by asynchronous acquisition or transport jitter.

To preserve traceability of data quality after preprocessing, a service-quality metadata component may be appended to the event structure.

A convenient representation is:

$$\hat{x}_i = \langle \tilde{v}_i, \tilde{s}_i, c_i, p_i, q_i \rangle,$$

where \tilde{v}_i and \tilde{s}_i denote corrected value and timestamp, and q_i is a compact quality descriptor (e.g., confidence, reconstruction flag, or synchronization reliability indicator). Such metadata is not merely auxiliary. It is required for consistent informativeness assessment because low-confidence events should not be interpreted identically to high-confidence events, even when their measured amplitudes are similar.

From a systems perspective, preprocessing serves two simultaneous functions: local improvement of data quality, and standardization of the event representation for later adaptive operators. This dual role is essential in resource-constrained IIoT nodes, where filtering and offloading decisions must be computed with low overhead and therefore rely on stable, compact, and semantically aligned inputs. Preprocessing is thus formalized as an integral control stage rather than a peripheral engineering step.

4.2 Event informativeness assessment. Adaptive local processing requires an explicit estimate of the informative value of each event (or event-derived unit), since resource-aware decisions cannot be based on amplitude thresholds alone in heterogeneous industrial streams.

A scalar informativeness functional is introduced to quantify the expected contribution of an event to

monitoring, diagnostics, or control support under the current process context.

This functional acts as the semantic control signal for subsequent filtering, aggregation, and prioritization rules.

Let x'_i denote the preprocessed event and H_i denote a compact local context (recent history of the same signal, same class, or same source). The informative value is defined as:

$$I_i = I(x'_i, H_i; \theta_I), \quad I_i \in [0, 1],$$

where θ_I is the parameter set of the informativeness model. The normalization $I_i \in [0, 1]$ is useful for constructing class-aware but comparable filtering rules and for coupling informativeness with distortion or compression constraints later in the pipeline.

A component-wise representation is adopted in order to preserve interpretability:

$$I_i = \sigma \left(\begin{array}{l} w_\Delta \Phi_\Delta(x'_i, H_i) + w_\Gamma \Phi_\Gamma(x'_i, H_i) + \\ + w_A \Phi_A(x'_i, H_i) + w_C \Phi_C(x'_i, H_i) \end{array} \right),$$

where Φ_Δ is a deviation component, Φ_Γ is a dynamics component, Φ_A is an anomaly-related component, Φ_C is a contextual component, $w_\Delta, w_\Gamma, w_A, w_C \geq 0$ are weights, and $\sigma(\cdot)$ is a bounded monotone mapping (e.g., logistic or clipped linear normalization). This decomposition reflects the fact that informative value in industrial systems is multi-factorial: a data point may be informative due to a large deviation, a rapid local trend, anomalous temporal behavior, or process context (e.g., class, priority, operational mode), even if no single component is dominant.

The deviation component Φ_Δ quantifies the distance from an expected or baseline state and may be defined using a class- or source-specific reference:

$$\Phi_\Delta(x'_i, H_i) = \rho_\Delta \left(\tilde{v}_i, \bar{v}_{c_i}(H_i) \right),$$

where $\bar{v}_{c_i}(H_i)$ is a local baseline estimate and ρ_Δ is a normalized discrepancy measure. The dynamics component Φ_Γ captures local rate-of-change, trend persistence, or short-term volatility:

$$\Phi_\Gamma(x'_i, H_i) = \rho_\Gamma(\tilde{v}_i, H_i).$$

The anomaly component Φ_A quantifies the degree of abnormality relative to the expected temporal or multivariate pattern and may be realized by lightweight local scoring procedures compatible with end-device constraints. The contextual component Φ_C embeds process semantics and operational relevance through class labels, priority levels, and optional quality metadata q_i :

$$\Phi_C(x'_i, H_i) = \rho_C(c_i, p_i, q_i, \zeta_i),$$

where ζ_i denotes optional context flags (e.g., current machine state or mode indicators).

The use of local history H_i is essential because informative value is generally nonstationary and path-dependent. The same measured value may be uninformative in a steady regime and highly informative during drift onset, recovery, or state transition. The history term enables local adaptation without requiring heavy cloud-side analytics and allows the end device to preserve early indicators of abnormal behavior even under constrained resources. The resulting I_i should therefore be interpreted as a context-conditioned utility estimate for resource-aware processing, rather than as a static anomaly score.

4.3 Adaptive filtering policy. Filtering is formalized as a state-dependent admissibility mechanism that determines whether an event is discarded, retained, or explicitly marked for protected handling in the downstream pipeline. The filtering layer does not perform full routing or offloading decisions; instead, it implements the first stage of resource-aware selection by controlling admission into subsequent local transformations. This separation is important for analytical clarity because it decouples semantic screening from action-level execution decisions while preserving a consistent state-based control structure.

Let $w(t_i)$ be the resource-state vector at the arrival time of event x'_i , and let I_i be its informative value. A filtering action variable is defined as:

$$m_i = F_{\theta_{\text{filt}}}(x'_i, I_i, w(t_i)),$$

with

$$m_i \in \{\text{drop}, \text{pass}, \text{mark}\}.$$

The label `drop` suppresses the event from the local processing chain, `pass` forwards it to the next stage under standard handling, and `mark` denotes protected or prioritized retention (e.g., for safety-critical, audit-relevant, or diagnostically significant events). This ternary formulation is preferable to a binary gate because it preserves the ability to encode policy-level exceptions and regulatory obligations without overloading the offloading stage.

A baseline threshold form is introduced as:

$$m_i = \begin{cases} \text{mark}, & I_i \geq \tau_{hi}(c_i, w(t_i)), \\ \text{pass}, & \tau_{lo}(c_i, w(t_i)) \leq I_i < \tau_{hi}(c_i, w(t_i)), \\ \text{drop}, & I_i < \tau_{lo}(c_i, w(t_i)). \end{cases}$$

The thresholds τ_{lo} and τ_{hi} are class-aware and state-dependent. Their dependence on $w(t)$ is the key adaptive mechanism that couples semantic relevance with current resource feasibility. Under queue growth, CPU saturation, or energy depletion, the thresholds are increased for low-priority classes, thereby reducing admitted load. Under favorable conditions, the thresholds are relaxed to preserve more context and improve observability (Fig. 4).

A convenient affine adaptation law for the lower threshold is:

$$\tau_{lo}(c_i, w) = \tau_{0,c_i} + \beta_Q^{(c_i)} \hat{Q} + \beta_U^{(c_i)} \hat{U}_{CPU} + \beta_S^{(c_i)} (1 - SoC) + \beta_R^{(c_i)} (1 - \hat{R}_{ch}),$$

where hatted variables denote normalized state components and τ_{0,c_i} is the nominal class-dependent

threshold. The upper threshold $\tau_{hi}(c_i, w)$ may be adapted analogously, with stricter protection for critical classes. This parameterization preserves interpretability and enables direct tuning of sensitivity to queue pressure, computational load, energy reserve, and communication conditions.

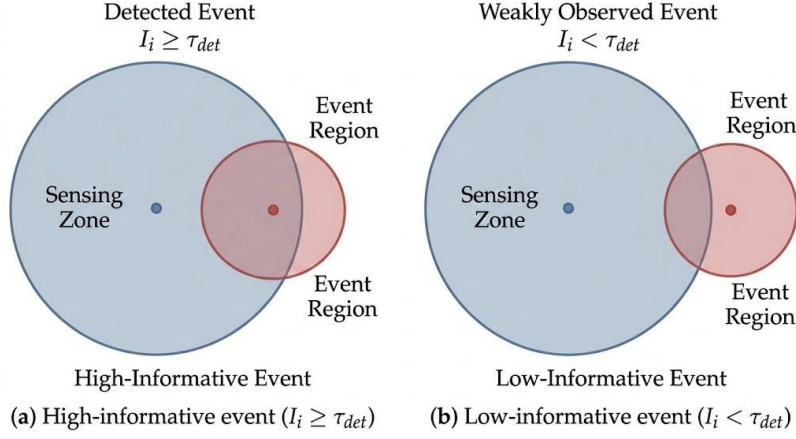


Fig. 4. Event classification based on informativeness assessment and adaptive detection thresholds

A probabilistic filtering variant is also admissible when deterministic thresholding produces undesirable switching near boundary regimes. In that case, the passing probability may be defined as:

$$\Pr(m_i \neq \text{drop} | x_i', w(t_i)) = \sigma\left(\kappa_{c_i} (I_i - \tau_{lo}(c_i, w(t_i)))\right),$$

with a steepness parameter κ_{c_i} . Such a formulation reduces chattering and improves robustness under noisy informativeness estimates while preserving the same state-coupled control logic.

Industrial operation typically imposes a minimum telemetry continuity requirement, especially for traceability and baseline monitoring. To account for this, a protected-rate constraint is introduced:

$$\lambda_{keep}^{(c)}(t) \geq \lambda_{\min}^{(c)}, \quad c \in \mathcal{C}_{prot},$$

where $\lambda_{keep}^{(c)}(t)$ is the retained rate of class c after filtering and \mathcal{C}_{prot} is the set of protected classes. This constraint prevents adaptive filtering from collapsing observability under overload and ensures that resource-aware operation remains compatible with industrial monitoring obligations.

4.4 Aggregation model. Aggregation is formalized as a controlled transformation that reduces event volume and representation granularity while preserving task-relevant semantics. In resource-constrained IIoT nodes, aggregation serves as a primary mechanism for communication-load reduction, but it also changes the temporal and structural properties of the data. For this reason, aggregation is treated as a parameterized operator embedded in the same adaptive framework as filtering and compression rather than as an independent post-processing utility.

Let \mathcal{W}_r denote an aggregation support set (window, group, or event collection) indexed by r .

The aggregated output unit is defined as:

$$z_r = A_{\theta_{agg}}(\mathcal{W}_r),$$

where θ_{agg} controls the aggregation mode, window size, class-specific rules, and semantic grouping criteria. The support set \mathcal{W}_r may be formed in several ways depending on workload characteristics and process requirements.

In temporal aggregation, the support set is a time window:

$$\mathcal{W}_r = \{x_i' | \tilde{s}_i \in [T_r, T_r + \Delta T_r)\},$$

and the output z_r contains compact descriptors of the events admitted in the window. The parameter ΔT_r may be fixed or state-adaptive. Increasing ΔT_r generally improves compression of traffic and reduces transmission frequency, but also smooths short transients and may delay the visibility of anomalies. The adaptive choice of ΔT_r therefore constitutes a direct trade-off between traffic reduction and temporal fidelity.

In event-driven aggregation, the support set is formed by event count, semantic trigger, or process condition rather than by wall-clock time. A generic representation is:

$$\mathcal{W}_r = \{x_i' | g_{evt}(x_i', H_i, w(t_i)) = 1\},$$

where g_{evt} is a grouping predicate. This form is useful for industrial streams with bursty alarms, mode transitions, or nonuniform sampling behavior, where time-window aggregation may combine semantically unrelated events. Event-driven grouping preserves process semantics more effectively and is especially valuable when local resources permit modest additional grouping logic.

When multiple local sources or channels are available at the same end node, spatio-temporal or semantic aggregation may be introduced. Let $x_i^{(n)}$ denote an event from source $n \in \mathcal{N}$. A multi-source support set may be defined as:

$$\mathcal{W}_r = \left\{ x_i^{(n)} \mid n \in \mathcal{N}_r, \tilde{s}_i \in [T_r, T_r + \Delta T_r), \chi(x_i^{(n)}) = 1 \right\},$$

where $\chi(\cdot)$ is a semantic compatibility predicate (e.g., same subsystem, same event class, same process stage). This formulation enables compact local summaries that preserve subsystem-level context and can reduce redundant upstream traffic from correlated sources.

The reduction effect of aggregation is characterized by a coefficient:

$$k_A = \frac{N_{in}}{N_{agg}},$$

where N_{in} is the number of admitted events and N_{agg} is the number of aggregated units over the observation horizon. A bitrate-oriented form may be used analogously. The parameter k_A is informative but not sufficient: high reduction alone does not imply acceptable performance if informative transients are suppressed. Aggregation quality must therefore be evaluated jointly with informativeness preservation and delay constraints introduced in the overall criteria vector.

4.5 Adaptive compression model. Compression and encoding are modeled as controlled transformations of aggregated or pre-aggregated representations into compact transmissible units. Their function is to reduce communication cost while preserving sufficient fidelity for local analysis and upper-layer processing. In the present framework, compression is not treated as a standalone communication primitive; it is integrated into the adaptive processing chain because the selected compression mode directly affects informativeness, downstream task quality, and the feasibility of offloading under channel constraints.

Let z_r denote an aggregated representation (or, in special cases, a retained event-level representation). The encoding process is defined by:

$$b_r = E_{\theta_{cmp}}(z_r),$$

where $E_{\theta_{cmp}}$ is the encoder and b_r is the encoded packet/bitstream. The corresponding decoded representation is:

$$\hat{z}_r = D_{\theta_{cmp}}(b_r).$$

The parameter block θ_{cmp} determines the encoding mode, quantization resolution, entropy-coding settings, and class-dependent compression profile. This abstraction remains implementation-neutral while explicitly preserving the control interface required for optimization.

Compression quality is characterized by two coupled indicators: bitrate (or code length) and distortion.

Let $|b_r|$ denote the encoded size. A local distortion measure is defined as:

$$d_r = d(z_r, \hat{z}_r),$$

where $d(\cdot, \cdot)$ may represent numerical reconstruction error, feature-space deviation, or task-oriented discrepancy depending on the representation type. The admissibility of a compression mode is restricted by class- or task-dependent distortion limits:

$$d_r \leq d_{\max}(c_r, p_r),$$

where c_r and p_r are the class and priority metadata inherited by the aggregated unit.

This constraint enforces differentiated protection of high-priority information and prevents excessive compression of diagnostically sensitive content.

A rate-informativeness trade-off can be represented through an informativeness loss functional $\ell_I^{(r)} = \mathcal{L}_I(z_r, \hat{z}_r)$, with a joint admissibility condition $\ell_I^{(r)} \leq \ell_{I, \max}(c_r, p_r)$. This form is particularly useful when classical numerical distortion does not adequately reflect analytical utility (e.g., when small waveform distortions erase weak fault precursors). The compression stage is therefore constrained not only by reconstruction fidelity but also by the preservation of informative content relevant to downstream diagnostics or control support.

Resource-aware encoding mode selection is represented by a policy:

$$\theta_{cmp}(t) = \Gamma_{cmp}(w(t), \bar{I}_r, c_r, p_r),$$

where \bar{I} is the aggregate informative value associated with z_r . Under low channel capacity, the policy may increase compression strength for low-priority units while preserving low-distortion modes for high-value units. Under high local CPU load, lightweight encoding modes may be preferred even at the cost of moderate bitrate increase.

This state-coupled formulation preserves consistency with the overall adaptive logic of the processing chain.

4.6 Feature representation for downstream processing. Feature formation is formalized as a transformation that maps retained or aggregated data into compact task-oriented descriptors suitable for local decision support and/or transmission to upper computational layers. The primary objective is not only dimensionality reduction, but the preservation of operationally relevant structure in a representation that is computationally efficient to generate, store, and transmit. This stage provides the bridge between data reduction and actionable processing, especially when direct transmission of raw or weakly processed events is impractical.

Let u_r denote the input to the feature stage, where u_r may be z_r , \hat{z}_r , or a filtered event-level representation depending on the active processing path. The feature vector is defined as:

$$y_r = \phi_{\theta_{feat}}(u_r, H_r),$$

where $\phi_{\theta_{feat}}$ is the feature extraction operator, H_r is a local contextual history, and θ_{feat} controls the feature set and computational profile. The resulting $y_r \in \mathbb{R}^{d_r}$ is a compact representation of the local process state or event group, with dimension d_r selected according to resource and task constraints.

A structured feature decomposition may be written as:

$$y_r = [y_r^{(stat)}, y_r^{(dyn)}, y_r^{(anom)}, y_r^{(ctx)}],$$

where $y_r^{(stat)}$ contains statistical descriptors (location, spread, robust moments), $y_r^{(dyn)}$ captures dynamics and trend information, $y_r^{(anom)}$ contains local anomaly indicators or residual scores, and $y_r^{(ctx)}$ encodes contextual metadata (class, priority, quality/confidence, or source group). This decomposition aligns with the informativeness model and ensures semantic continuity between filtering and feature-based downstream processing.

When further compactness is required, a dimensionality reduction operator may be applied:

$$\tilde{y}_r = R_{\theta_{red}}(y_r), \dim(\tilde{y}_r) = d_r^* \ll d_r.$$

The operator $R_{\theta_{red}}$ may represent a linear projection, a structured feature selector, or another lightweight reduction scheme compatible with end-device constraints. The admissibility of reduction is again governed by informativeness preservation and task adequacy rather than by dimensionality alone. In practice, low-dimensional representations are preferred only if they retain sufficient sensitivity to critical deviations and process transitions.

For interaction with local analysis modules or upper layers, a compact task packet representation is introduced:

$$\xi_r = \langle \tilde{y}_r, c_r, p_r, \eta_r \rangle,$$

where η_r contains auxiliary metadata such as confidence, aggregation profile, compression mode, and provenance flags. This packet structure is suitable both for local lightweight inference and for offloading to fog/cloud resources, because it preserves the minimum semantic and quality information needed for informed downstream interpretation.

The computational complexity of feature formation must remain compatible with end-device resource limits. Accordingly, the feature operator is constrained by a local budget condition of the form:

$$C_\phi(u_r, \theta_{feat}) \leq C_{\max}(w(t_r)),$$

where C_ϕ denotes the computational cost of feature extraction and C_{\max} is the state-dependent allowable budget. This explicitly embeds feature formation into the same resource-aware control logic as filtering,

aggregation, and compression and ensures that compact representation learning does not destabilize local queuing or energy behavior.

The resulting pipeline of preprocessing, informative value assessment, filtering, aggregation, compression, and feature formation provides a mathematically consistent local transformation chain for heterogeneous IIoT events. The outputs of this chain retain semantic relevance and resource-aware compactness, thereby forming an appropriate basis for subsequent local analysis and for adaptive allocation of processing between end, fog, and cloud layers.

4.7 Decision and offloading interface. The output of the adaptive local processing chain is a compact, semantically annotated representation that serves as the input to the decision layer responsible for local execution or transfer to upper computational tiers. Let ξ_r denote the processed unit produced after preprocessing, informative-value assessment, filtering, aggregation, compression, and feature formation. Depending on the active transformation path, ξ_r may correspond to an event-level descriptor or an aggregated task-oriented packet, but in all cases, it is assumed to preserve the minimum information required for downstream interpretation:

$$\xi_r = \langle \tilde{y}_r, \bar{I}_r, c_r, p_r, \eta_r \rangle,$$

where ξ_r is the compact representation, \bar{I}_r is the associated informative value (or aggregated informative score), c_r and p_r are class and priority metadata, and η_r contains auxiliary descriptors such as confidence, provenance, aggregation profile, and compression mode.

Decision-making is defined over the pair $(\xi_r, w(t_r))$, where $w(t_r)$ is the resource-state vector of the end device at the decision epoch t_r . This pairing is fundamental because the same processed unit may require different handling under different operating conditions. In particular, high-informativeness units may be processed locally under sufficient computational capacity, deferred under transient overload, or transferred to fog/cloud resources when local execution is infeasible but communication conditions are acceptable. Conversely, low-informativeness units may be retained locally in benign regimes but suppressed or heavily compressed under resource pressure. The decision interface is therefore state-dependent by construction and cannot be reduced to a static routing rule. A compact decision mapping is introduced in the form:

$$a_r^* = \pi_{dec}(\xi_r, w(t_r)),$$

where a_r^* is the selected handling action and π_{dec} is the decision policy. At the interface level, the internal structure of π_{dec} is left abstract; only its inputs and output contract are fixed. This abstraction is sufficient to preserve formal continuity of the pipeline while avoiding premature commitment to a specific optimization or control mechanism. The action variable a_r^* is assumed to belong to an admissible set that includes local

handling, deferred handling, and offloading to upper layers, subject to resource and operational constraints.

Feasibility is defined through a constraint-filtered action set:

$$\mathcal{A}_r^{feas} = \{a : g_j(a, \xi_r, w(t_r)) \leq 0, j = 1, \dots, J\},$$

where $g_j(\cdot)$ encodes resource, timing, communication, and policy constraints. This formulation ensures that the output of the local processing chain is not interpreted as an unconstrained optimization object; any downstream decision remains bounded by queue stability conditions, CPU and memory limits, energy constraints, channel availability, and class-dependent service requirements. In addition, protected telemetry continuity constraints remain active at the decision interface, preventing irreversible loss of critical units even in degraded regimes.

The interface representation also preserves the coupling between local transformation quality and subsequent offloading decisions. In particular, metadata η_r and informative value \bar{I}_r provide the decision layer with explicit information about how the unit was generated (e.g., degree of aggregation, compression mode, confidence level), which is necessary for evaluating suitability of local analysis versus transmission. Without this interface contract, the decision layer would operate on opaque payloads and would be unable to correctly account for representation quality, uncertainty, or the expected utility of further processing at fog/cloud levels. A horizon-level view is obtained by considering the sequence:

$$\Xi_H = \{\xi_r \mid t_r \in [t_0, t_0 + H]\},$$

which forms the input stream to the decision layer over observation horizon H . The local processing stages defined above thus provide not only data reduction, but also a normalized and decision-ready representation of heterogeneous IIoT traffic. This establishes a mathematically explicit bridge from adaptive event transformation to constrained decision-making for local processing and offloading, while preserving sufficient modularity for alternative policy realizations in subsequent developments.

5. Discussion

The proposed formalization provides a coherent methodological basis for adaptive information processing on IIoT end devices by explicitly separating the state-level description of the system from the operator-level logic of local data transformation. This separation is analytically advantageous because it avoids conflating architectural assumptions, resource constraints, and processing mechanisms within a single undifferentiated model. The event-stream representation, queue dynamics, and resource-state vector define the system conditions under which adaptation must operate, whereas the local processing chain comprising preprocessing, informative-value assessment, adaptive filtering, aggregation, compression, and feature formation defines the controllable transformations through which those conditions and the resulting performance criteria can be influenced. Such a decomposition is not only structurally

clear, but also compatible with journal-level formalization, since it preserves explicit interfaces between model components and supports progressive extension toward more detailed control and optimization layers.

A central methodological contribution lies in the treatment of local processing as a sequence of parameterized operators rather than as a fixed implementation pipeline. This operator-centric view preserves generality across heterogeneous IIoT domains, where sensing modalities, event semantics, communication regimes, and resource budgets differ substantially. At the same time, it enables formal reasoning about inter-stage dependencies, which is essential in industrial environments because any design choice at one stage propagates to subsequent stages and modifies their feasible operating region. For example, stricter filtering reduces downstream load but changes aggregation support sets and may alter the admissibility of compression modes; larger aggregation windows reduce communication volume but affect temporal fidelity and the interpretability of derived features; more expressive feature sets improve downstream analytical utility but consume additional local computational budget. The model therefore captures local processing not as a collection of independent utilities, but as a coordinated mechanism for controlled information shaping under constrained resources.

The introduction of a formal interface to decision-making and offloading further strengthens the internal coherence of the framework. By representing the output of the local pipeline as a compact, semantically annotated unit ξ_r and defining decision-making over the pair $(\xi_r, w(t_r))$, the formulation establishes a mathematically explicit contract between local transformations and subsequent execution or routing decisions. This is particularly important for preventing a common limitation of pipeline-only models, in which downstream decision logic is left implicit and therefore cannot be rigorously analyzed. Even though the internal structure of the decision policy is not fully specified in the present version, the interface itself ensures that future local-processing/offloading policies can be designed over a well-defined input space that already contains informative-value estimates, class and priority metadata, and representation-quality descriptors. In this sense, the framework remains modular without becoming under-specified.

The developed formalism also makes the key trade-offs of adaptive local processing explicit. The informative-value functional and adaptive filtering rules demonstrate that resource-aware event selection cannot be reduced to simple amplitude thresholding in heterogeneous industrial streams. Increasing filtering aggressiveness may reduce queue pressure, CPU usage, and communication load, but it simultaneously increases the risk of suppressing weak, early-stage, or context-dependent indicators of abnormal process behavior. The use of class-aware and state-dependent thresholds alleviates this tension by protecting critical traffic and adjusting admissibility according to resource conditions, yet it does not eliminate the underlying conflict. This is an expected and methodologically correct outcome: the purpose of the model is not to remove trade-offs at the formal level, but

to expose them in a parameterized form suitable for controlled synthesis and future empirical evaluation.

Aggregation and compression introduce additional trade-offs that are equally important from an industrial perspective. Temporal aggregation improves traffic efficiency and can stabilize local operation under bursty arrivals, but it smooths transient structures and may delay the visibility of short-duration anomalies. Event-driven and semantic aggregation reduce this risk by preserving process-aligned grouping logic, but they typically require more local state tracking and slightly higher computational complexity. Compression, in turn, cannot be treated solely as a bitrate minimization tool, because numerical distortion and informative-value loss are not equivalent in many IIoT tasks. A small reconstruction error may be acceptable for slowly varying telemetry, yet the same distortion magnitude may be unacceptable if it degrades weak fault signatures, threshold-crossing events, or subtle trend changes relevant to diagnostics. The explicit distinction between distortion bounds and informative-value preservation constraints is therefore a significant conceptual advantage of the proposed formulation, as it aligns communication-oriented reduction with task-oriented utility preservation.

Feature formation occupies a particularly important role in reconciling compact representation with downstream utility. The formalization adopted here treats feature extraction as a resource-constrained transformation rather than a purely analytical enhancement step, which is appropriate for end-device deployment conditions. Richer feature representations may improve local decision support and reduce the need for raw-data transmission, but they also increase local computational cost and may adversely affect queue stability or energy consumption. Embedding feature formation into the same state-aware control logic as filtering, aggregation, and compression ensures that compactness, informativeness, and computational feasibility are jointly considered. This reinforces the interpretation of the local pipeline as a coordinated information-shaping mechanism, not merely a traffic-reduction chain.

From an architectural standpoint, the framework is well aligned with cloud-fog-edge processing principles. The local chain does not attempt to replicate cloud-scale analytics on the end device; instead, it constructs a decision-ready, semantically structured representation that preserves relevant process information under strict local constraints. This distinction is essential in industrial distributed systems, where the end device must support low-latency responsiveness and bandwidth-aware data preparation, while fog and cloud layers remain responsible for more computationally demanding analytics and broader system-level coordination. The explicit interface based on ξ_r supports this partitioning by defining a stable boundary between local adaptive processing and higher-tier computation. It also improves extensibility, since different deployments may implement different decision policies, communication mechanisms, or fog/cloud allocation strategies while preserving the same local operator formalization and system-state model.

The proposed structure also has implications for observability and traceability in industrial environments. Traffic-centric edge reduction methods often improve resource metrics at the cost of diminished forensic visibility and poorer post-event diagnostics, especially when low-intensity contextual signals are discarded. The present formalization addresses this issue through class-aware informativeness assessment, protected filtering marks, continuity constraints, and metadata propagation across the processing chain. These mechanisms do not guarantee perfect retention of all diagnostically relevant information, but they provide a principled basis for balancing resource efficiency with monitoring obligations, auditability, and incident reconstruction requirements. This balance is particularly important in industrial settings where compliance and traceability may be as critical as real-time control performance.

Several practical considerations follow directly from the selected abstractions. The effectiveness of adaptive filtering and informative-value assessment depends on the stability of preprocessing, synchronization quality, and the reliability of local baselines or context windows. Errors introduced at the preprocessing stage can propagate into informativeness estimates and lead to systematic misclassification of events, especially under nonstationary operating conditions. The inclusion of quality metadata mitigates this risk, but practical deployment still requires careful calibration of preprocessing parameters and monitoring of data-quality indicators. In addition, state-based adaptation assumes sufficiently accurate and timely estimates of queue occupancy, CPU load, memory pressure, energy state, and communication conditions. While some of these quantities are readily available on-device, channel-state indicators may be noisy or delayed in real deployments, which can induce threshold oscillations or unstable compression-mode switching unless smoothing, hysteresis, or bounded adaptation rates are used. The formal model remains compatible with such mechanisms, but their explicit design belongs to implementation-specific realizations.

At the same time, the abstraction level of the framework makes it suitable for deployment on constrained embedded platforms because each stage is parameterized and can be instantiated at different complexity levels.

Lightweight variants of preprocessing, anomaly scoring, aggregation logic, or feature extraction may be used when deterministic timing or memory limits dominate, while richer variants can be introduced on more capable nodes without changing the overall mathematical structure. This scalability across device capabilities is an important practical advantage, since industrial IIoT environments often contain heterogeneous end devices with different computational and energy budgets.

The principal limitation of the present contribution is the absence of empirical validation. The current text establishes the formal and architectural foundation of adaptive local processing, but it does not provide simulation results, benchmark comparisons, or deployment measurements. Consequently, no

quantitative claims are made regarding delay reduction, energy savings, Pareto improvements, or robustness margins. This is a limitation of evidence rather than of formal completeness, and it implies that the framework should be interpreted as a methodological basis for subsequent evaluation rather than as a finalized performance claim. A second limitation is the abstract treatment of the decision policy behind the interface mapping $\pi_{dec}(\xi_r, w(t_r))$. This abstraction is justified by the present focus on local processing logic, yet it postpones explicit analysis of decision-rule structure, local/offloading action selection, and the interaction between decision policies and operator parameters under nonstationary workloads. A third limitation concerns uncertainty propagation. Although quality metadata and state dependence are included, the framework does not yet provide a full probabilistic treatment of uncertainty across preprocessing, informativeness estimation, aggregation, compression, and feature formation, which would be valuable for safety-critical industrial scenarios requiring risk-bounded adaptation.

These limitations define a clear path for further development. Future work should instantiate the decision interface with a constrained policy for local processing and offloading, examine deterministic and stochastic decision variants, and study their interaction with operator-level parameters under realistic workload dynamics. Benchmark-oriented evaluation across representative industrial scenarios with different class compositions, deadline profiles, and communication conditions is also necessary to identify stable operating regions and quantify trade-offs in the criteria space. In addition, extending the framework toward uncertainty-aware modeling would improve its applicability in safety-critical domains by enabling confidence-sensitive filtering, robust compression control, and risk-aware downstream decision-making. Despite these open directions, the present formalization already provides a consistent and sufficiently general foundation for adaptive IIoT end-device processing in cloud-fog-edge environments, with its main scientific value residing in the explicit coupling of event semantics, resource-state awareness, and operator-level data transformation within a unified mathematical framework.

Conclusions

The paper developed a formal methodological framework for adaptive information processing on IIoT end devices in cloud-fog-edge environments, with emphasis on the internal logic of local data transformation under resource constraints. The proposed formulation integrates a state-level system description and an operator-level processing chain within a single mathematical structure.

At the system level, the framework combines an event-based representation of heterogeneous industrial data, a queue-oriented model of local processing flow, a resource-state vector capturing computational, energy, and communication conditions, and a multi-criteria performance representation that jointly accounts for delay, loss, resource consumption, communication load, and informative value preservation. This system model

provides the analytical basis for describing adaptive local processing not as an isolated implementation pipeline, but as a controllable component of distributed IIoT information management.

A core contribution of the paper is the operator-level formalization of the local processing stages. Preprocessing and normalization were represented as a parameterized transformation that ensures structural consistency and quality-aware input preparation.

Informative-value assessment was introduced as a context-dependent functional that quantifies the relevance of events for subsequent monitoring and diagnostics.

Adaptive filtering was formalized through state-dependent admissibility rules that couple semantic relevance with resource feasibility.

Aggregation and compression were modeled as controlled information-reduction operators with explicit consideration of fidelity and informative-value loss, while feature formation was defined as a resource-constrained transformation into compact task-oriented representations. Taken together, these stages form a mathematically explicit local processing chain capable of shaping heterogeneous event streams into semantically meaningful and resource-efficient units suitable for further handling.

An additional contribution of the paper is the introduction of a formal interface between local adaptive processing and subsequent decision-making for local execution or offloading. By defining a processed unit that carries compact features, informative-value estimates, and semantic/quality metadata, the framework establishes a consistent input contract for downstream decision policies without fixing a specific decision mechanism within the present scope. This interface preserves modularity while ensuring formal continuity between local transformations and higher-level allocation of computation across end, fog, and cloud layers. As a result, the proposed structure remains suitable for multiple decision-policy realizations and deployment contexts, including systems with different latency requirements, communication regimes, and device capabilities.

The main scientific value of the presented framework lies in the explicit coupling of event semantics, resource-state awareness, and operator-level transformations in a unified formal model. This coupling is essential for industrial IIoT scenarios, where improvements in one criterion (e.g., reduced traffic or lower local load) may degrade others (e.g., observability, diagnostic fidelity, or timeliness). The proposed formulation does not eliminate these trade-offs; rather, it exposes them in a structured, parameterized form that supports rigorous analysis and subsequent synthesis. In this sense, the framework provides a foundation for future optimization and control procedures that can operate on a well-defined mathematical representation of local adaptive processing.

The paper is intentionally limited to formalization and does not include simulation-based or deployment-based validation. Consequently, no quantitative performance claims are made in terms of delay reduction,

energy savings, or Pareto efficiency gains. In addition, the downstream decision policy for local processing and offloading is left abstract behind the interface mapping, and uncertainty propagation across the processing stages is not yet modeled probabilistically. These limitations define the immediate directions for further research. Future work should instantiate the decision layer as a constrained policy, evaluate deterministic and stochastic action-selection mechanisms, analyze parameter sensitivity under nonstationary industrial workloads, and perform benchmark-oriented validation across representative IIoT scenarios. Extending the framework toward uncertainty-aware and risk-bounded adaptation is also a relevant next step for safety-critical applications.

Overall, the proposed formalization establishes a consistent and extensible basis for adaptive end-device

information processing in distributed IIoT systems. Its structure is sufficiently general for domain-specific instantiation and sufficiently rigorous to support subsequent development of decision policies, optimization methods, and empirical validation studies in cloud-fog-edge architectures.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

REFERENCES

1. Alsadie, D. (2024), "Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: challenges and prospects", *PeerJ Computer Science*, vol. 10, e2128, doi: <https://doi.org/10.7717/PEERJ-CS.2128>
2. Kaur, G., Balyan, V., and Gupta, S.H. (2025), "Nature inspired optimization of IoT network for delay resistant and energy efficient applications", *Scientific Reports*, vol. 15(1), 9902, doi: <https://doi.org/10.1038/s41598-025-95138-z>
3. Yu, J., Hou, K., Zhang, H., Kostic B. Yang, M., and Nazif, H. (2025), "A new energy-aware resources scheduling method for mobile internet of things using a hybrid optimisation algorithm", *International Journal of Mobile Communications*, vol. 25(2), pp. 176–207, doi: <https://doi.org/10.1504/IJMC.2025.144192>
4. Schulz, A.S. (2023), "User Interactions with Internet of Things (IoT) Devices in Shared Domestic Spaces", *ACM International Conference Proceeding Series*, pp. 577–579, doi: <https://doi.org/10.1145/3626705.3632615>
5. Gamboa, A., Villazón, A., Meneses, A., Ormachea, O., and Orellana, R. (2024), "Altitude's Impact on Photovoltaic Efficiency: An IoT-Enabled Geographically Distributed Remote Laboratory", *Lecture Notes in Networks and Systems*, 1028 LNNS, pp. 133–144, doi: https://doi.org/10.1007/978-3-031-61905-2_14
6. Liu, X., Lu, D., Zhang, A., Liu, Q., and Jiang, G. (2022), "Data-Driven Machine Learning in Environmental Pollution: Gains and Problems", *Environmental Science and Technology*, vol. 56(4), pp. 2124–2133, doi: <https://doi.org/10.1021/acs.est.1c06157>
7. Kuchuk, H., Kalinin, Y., Dotsenko, N., Chumachenko, I. and Pakhomov, Y. (2024), "Decomposition of integrated high-density IoT data flow", *Advanced Information Systems*, vol. 8, no. 3, pp. 77–84, doi: <https://doi.org/10.20998/2522-9052.2024.3.09>
8. Vaiyapuri, T., Parvathy, V.S., Manikandan, V., Krishnaraj N., Gupta, D., and Shankar, K. (2022), "A Novel Hybrid Optimization for Cluster-Based Routing Protocol in Information-Centric Wireless Sensor Networks for IoT Based Mobile Edge Computing", *Wireless Personal Communications*, vol. 127(1), pp. 39–62, doi: <https://doi.org/10.1007/s11277-021-08088-w>
9. Pardo, C., Wei, R., and Ivens, B.S. (2022), "Integrating the business networks and internet of things perspectives: A system of systems (SoS) approach for industrial markets", *Industrial Marketing Management*, vol. 104, pp. 258–275, doi: <https://doi.org/10.1016/j.indmarman.2022.04.012>
10. Singh, S.P., Kumar, N., Kumar, G., Balusamy, B., Bashir, A.K., and Al Dabel, M.M. (2025), "Enhancing Quality of Service in IoT-WSN through Edge-Enabled Multi-Objective Optimization", *IEEE Transactions on Consumer Electronics*, vol. 71, is. 2, doi: <https://doi.org/10.1109/TCE.2025.3526992>
11. Kuchuk, H., and Malokhvii, E. (2024), "Integration of iot with Cloud, Fog and Edge computing: a review", *Advanced Information Systems*, vol. 8, no. 2, pp. 65–78, doi: <https://doi.org/10.20998/2522-9052.2024.2.08>
12. Qayyum, T., Trabelsi, Z., Waqar Malik, A., and Hayawi, K. (2022), "Mobility-aware hierarchical fog computing framework for Industrial Internet of Things", *Journal of Cloud Computing*, vol. 11(1), doi: <https://doi.org/10.1186/s13677-022-00345-y>
13. Muñoz, L.A., Berná Martínez, J.V., Asensi, C.C., and Pastor, D.S. (2025), "Research Notes: Design of a Distributed and Highly Scalable Fog Architecture for Heterogeneous IoT Infrastructures", *Int. Journal of Software Engineering and Knowledge Eng.*, vol. 35(2), pp. 195–215, doi: <https://doi.org/10.1142/S0218194025430016>
14. Oleh, Z., and Oleksii, L. (2024), "Architecture and IoT security systems based on Fog computing", *Innovative Technologies and Scientific Solutions for Industries*, vol. 2024(1(27)), pp. 54–66, doi: <https://doi.org/10.30837/ITSSI.2024.27.054>
15. Yan, M. (2024), "Receive wireless sensor data through IoT gateway using web client based on border gateway protocol", *Heliyon*, vol. 10(11), e31625, doi: <https://doi.org/10.1016/j.heliyon.2024.e31625>
16. Kuchuk, H., Husieva, Y., Novoselov, S., Lysytsia, D., Krykhovetskyi, H. (2025), "Load Balancing of the layers Iot Fog-Cloud support network", *Advanced Information Systems*, vol. 9, no. 1, pp. 91–98, doi: <https://doi.org/10.20998/2522-9052.2025.1.11>
17. Kuchuk, H., Chumachenko, I., Marchenko, N., Kuchuk, N., and Lysytsia, D. (2025), "Method for calculating the number of IOT sensors in environmental monitoring systems", *Advanced Information Systems*, vol. 9, no. 3, pp. 66–32, doi: <https://doi.org/10.20998/2522-9052.2025.3.08>
18. Abdullayeva, M.Y., Aghayev, B.S. and Yaqubov, R.V. (2024), "Problems of environmental pollution with microplastic waste and ways to solve them", *Bio Web of Conferences*, vol. 95, 02002, doi: <https://doi.org/10.1051/bioconf/20249502002>
19. Mani Kiran, C.V.N.S., Jagadeesh Babu, B., and Singh, M.K. (2023), "Study of Different Types of Smart Sensors for IoT Application Sensors", *Smart Innovation, Systems and Technologies*, vol. 290, pp. 101–107, doi: https://doi.org/10.1007/978-981-19-0108-9_11
20. Kuchuk, N., Kashkevich, S., Radchenko, V., Andrusenko, Y., and Kuchuk, H. (2024), "Applying edge computing in the execution IoT operative transactions", *Advanced Information Systems*, vol. 8, no. 4, pp. 49–59, doi: <https://doi.org/10.20998/2522-9052.2024.4.07>

21. Bhajantri, L.B., and Gangadharaiah, S. (2023), “Heuristic-Based Resource Allocation for Internet of Things in Gateway Centric Multi-layer Fog Computing”, *Lecture Notes in Networks and Systems*, vol. 516, pp. 567–579, doi: https://doi.org/10.1007/978-981-19-5221-0_54
22. Li, G., Liu, Y., Wu, J., Lin, D., and Zhao, Sh. (2019), “Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing”, *Sensors*, MDPI, vol. 19(9), doi: <https://doi.org/10.3390/s19092122>
23. Malokhvii, E. and Kuchuk, H. (2025), “Differential Evolution for Optimized Task Clustering and Scheduling in IoT Edge Computing”, *2025 IEEE 6th Khpi Week on Advanced Technology (Khpiweek 2025)*, doi: <https://doi.org/10.1109/KhPIWeek61436.2025.11288695>
24. Gomathi, B., Saravana Balaji, B., Krishna Kumar, V., Abouhawwash, M., Aljahdali, S., Masud, M., and Kuchuk, N. (2022), “Multi-Objective Optimization of Energy Aware Virtual Machine Placement in Cloud Data Center”, *Intelligent Automation and Soft Computing*, vol. 33(3), pp. 1771–1785, doi: <http://dx.doi.org/10.32604/iasc.2022.024052>
25. Amitu, D.M., Akol, R.N., and Serugunda, J. (2025), “Hybrid access control mechanism for massive machine type communications”, *Discover Internet of Things*, vol. 5(1), 30, doi: <https://doi.org/10.1007/s43926-025-00106-8>
26. Tzeng, S.-S., Lin, Y.-J., and Wang, S.-W. (2025), “Age of Information in IoT Devices with Integrated Heterogeneous Sensors under Slotted ALOHA”, *IEEE Sensors Journal*, doi: <https://doi.org/10.1109/JSEN.2025.3563452>
27. Petrovska, I., Kuchuk, H., Kuchuk, N., Mozhaiev, O., Pochebut, M. and Onishchenko, Yu. (2023), “Sequential Series-Based Prediction Model in Adaptive Cloud Resource Allocation for Data Processing and Security”, *2023 13th International Conference on Dependable Systems, Services and Technologies, DESSERT 2023*, 13–15 October, Athens, Greece, code 197136, doi: <https://doi.org/10.1109/DESSERT61349.2023.10416496>
28. Sharma, Sh. Saini H. (2019), “A novel four-tier architecture for delay aware scheduling and load balancing in fog environment”, *Sustainable Computing: Informatics and Systems*, vol. 24, doi: <https://doi.org/10.1016/j.suscom.2019.100355>
29. Khudov, H., Diakonov, O., Kuchuk, N., Maliuha, V., Furmanov, K., Mylashenko, I., Olshevskiy, Y., Stetsiv, S., Solomonenko, Y., and Yuzova, I. (2021), “Method for determining coordinates of airborne objects by radars with additional use of ads-b receivers”, *Eastern-European Journal of Enterprise Technologies*, vol. 4(9(112)), pp. 54–64, doi: <https://doi.org/10.15587/1729-4061.2021.238407>
30. Malik, U.M., Javed, M.A., Frnda, J., Rozhon, J., and Khan, W.U. (2022), “Efficient Matching-Based Parallel Task Offloading in IoT Networks”, *Sensors*, vol. 22, doi: <https://doi.org/10.3390/s22186906>
31. Liu, L., Chen, H., and Xu, Z. (2022), “SPMOO: A Multi-Objective Offloading Algorithm for Dependent Tasks in IoT Cloud-Edge-End Collaboration”, *Information*, vol. 13, 75, doi: <https://doi.org/10.3390/info13020075>
32. Ghenai, A., Kabouche, Y., and Dahmani, W. (2018), “Multi-user dynamic scheduling-based resource management for Internet of Things applications”, *International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, doi: <https://doi.org/10.1109/IINTEC.2018.8695308>
33. Wei, J.-Y., and Wu, J.-J. (2023), “Resource Allocation Algorithm in Industrial Internet of Things Based on Edge Computing”, *Dongbei Daxue Xuebao / Journal of Northeastern University*, vol. 44(8), doi: <https://doi.org/10.12068/j.issn.1005-3026.2023.08.002>
34. Zhang, Z. (2023), “A computing allocation strategy for Internet of things’ resources based on edge computing”, *International Journal of Distributed Sensor Networks*, vol. 17(12), doi: <https://doi.org/10.1177/15501477211064800>
35. Kovalenko, A., and Kuchuk, H. (2022), “Methods to Manage Data in Self-healing Systems”, *Studies in Systems, Decision and Control*, vol. 425, pp. 113–171, doi: https://doi.org/10.1007/978-3-030-96546-4_3
36. Pisching, M., Pessoa, M. A. O., Junqueira, F., Filho, D. J. S., and Miyagi, P. E. (2018), “An architecture based on RAMI 4.0 to discover equipment to process 168 operations required by products”, *Computers & Industrial Engineering*, vol. 125, pp. 574–591, doi: <https://doi.org/10.1016/j.cie.2017.12.029>
37. Fatlawi, A., and Al Dujaili, M.J. (2023), “Integrating the Internet of Things (IoT) and Cloud Computing Challenges and Solutions: A Review”, *AIP Conference Proceedings*, vol. 2977(1), 020067, doi: <http://dx.doi.org/10.1063/5.0181842>
38. Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Tymoshchuk, L., Onishchenko, Yu., Tulupov, V., Bykova, T., and Roh, V. (2025), “Devising a method for forming a stable mobile cluster of the internet of things fog layer”, *Eastern-European Journal of Enterprise Technologies*, 2025, vol. 1, no. 4(133), pp. 6–14, doi: <https://doi.org/10.15587/1729-4061.2025.322263>
39. Jamil, B. Shojafar, M., Ahmed, I., Ullah, A., Munir, K., and Ijaz, H. (2020), “A job scheduling algorithm for delay and performance optimization in fog computing”, *Concurrency and Computation: Practice and Experience*, vol. 32(7), doi: <https://doi.org/10.1002/cpe.5581>
40. Proietti Mattia, G., and Beraldi, R. (2023), “P2PFaaS: A framework for FaaS peer-to-peer scheduling and load balancing in Fog and Edge computing”, *SoftwareX*, vol. 21, doi: <https://www.sciencedirect.com/science/article/pii/S2352711022002084>
41. Lu, S., Wu, J., Wang, N., Duan, Y., Liu, H., Zhang, J., and Fang, J. (2023), “Resource provisioning in collaborative fog computing for multiple delay-sensitive users”, *Software – Practice and Experience*, vol. 53(2), pp. 243–262, doi: <https://doi.org/10.1002/spe.3000>
42. Drabech, Z., Douimi, M., and Zemmouri, E. (2024), “A Markov random field model for change points detection”, *Journal of Computational Science*, vol. 83, 102429, doi: <https://doi.org/10.1016/j.jocs.2024.102429>
43. Mutambik, I. (2024), “An Entropy-Based Clustering Algorithm for Real-Time High-Dimensional IoT Data Streams”, *Sensors*, vol. 24(22), 7412, doi: <https://doi.org/10.3390/s24227412>
44. Petrovska I., Kuchuk, H., and Mozhaiev M. (2022), “Features of the distribution of computing resources in cloud systems”, *2022 IEEE 3rd KhPI Week on Advanced Technology, KhPI Week 2022 - Conference Proceedings*, 03-07 October 2022, doi: <https://doi.org/10.1109/KhPIWeek57572.2022.9916459>
45. Thomas, P., and Jose, D.V. (2023), “Towards Computation Offloading Approaches in IoT-Fog-Cloud Environment: Survey on Concepts, Architectures, Tools and Methodologies”, *Lecture Notes in Networks and Systems*, 613 LNNS, pp. 37–52, doi: https://doi.org/10.1007/978-981-19-9379-4_4
46. Emami Khansari, M., and Sharifian, S. (2024), “A scalable modified deep reinforcement learning algorithm for serverless IoT microservice composition infrastructure in fog layer”, *Future Generation Computer Systems*, vol. 153, pp. 206–221, doi: <https://doi.org/10.1016/j.future.2023.11.022>

47. Semenov, S., Mozhaiev, O., Kuchuk, N., Mozhaiev, M., Tiulieniev, S., Gnusov, Yu., Yevstrat, D., Chyrva, Y., Kuchuk, H. (2022), "Devising a procedure for defining the general criteria of abnormal behavior of a computer system based on the improved criterion of uniformity of input data samples", *Eastern-European Journal of Enterprise Technologies*, vol. 6(4-120), pp. 40–49, doi: <https://doi.org/10.15587/1729-4061.2022.269128>
48. Azimi, S., Pahl, C., and Shirvani, M.H. (2020), "Particle swarm optimization for performance management in multi-cluster IoT edge architectures", *CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science*, pp. 328–337, doi: <https://doi.org/10.5220/0009391203280337>
49. Simaiya, S., Shrivastava, A., and Keer, N.P. (2014), "IREM algorithm for improvement in performance of mobile ad hoc networks", *Proceedings - 2014 4th International Conference on Communication Systems and Network Technologies*, CSNT 2014, 6821402, doi: <https://doi.org/10.1109/CSNT.2014.62>
50. Qasim, M., and Sajid, M. (2024), "An efficient IoT task scheduling algorithm in cloud environment using modified Firefly algorithm", *International Journal of Information Technology (Singapore)*, vol. 17, pp. 179–188, doi: <https://doi.org/10.1007/s41870-024-01758-5>
51. Carvalho, D., Sullivan, D., Almeida, R., and Caminha, C. (2022), "A Machine Learning Approach to Solve the Network Overload Problem Caused by IoT Devices Spatially Tracked Indoors", *Journal of Sensor and Actuator Networks*, vol. 11(2), 29, doi: <https://doi.org/10.3390/jsan11020029>
52. Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Tymoshchyk, L., Lubentsov, A., Gnusov, Y., Klivets, S. and Kuleshov, A. (2025), "Devising a method for stabilizing control over a load on a cluster gateway in the internet of things edge layer", *Eastern-European Journal of Enterprise Technologies*, vol. 2(9 (134)), pp. 24–32, doi: <https://doi.org/10.15587/1729-4061.2025.326040>
53. Choi, J. (2018), "On Multichannel Random Access for Correlated Sources. *IEEE Transactions on Communications*", vol. 66(8), pp. 3444–3454, 8331969, doi: <https://doi.org/10.1109/TCOMM.2018.2823318>
54. Chetot, L., Egan, M., and Gorce, J.-M. (2023), "Active User Detection and Channel Estimation for Grant-Free Random Access with Gaussian Correlated Activity", *IEEE Vehicular Technology Conference*, 2023-June, 191756, doi: <https://doi.org/10.1109/VTC2023-Spring57618.2023.10199877>
55. Kuchuk, N., Mozhaiev, O., Semenov, S., Haichenko, A., Kuchuk, H., Tiulieniev, S., Mozhaiev, M., Davydov, V., Brusakova, O., and Gnusov, Y. (2023), "Devising a method for balancing the load on a territorially distributed foggy environment", *Eastern-European Journal of Enterprise Technologies*, vol. 1(4 (121)), pp. 48–55, doi: <https://doi.org/10.15587/1729-4061.2023.274177>
56. Alqasimi, A., Al Marzouqi, K., Alhammadi, A., Aljasmii, A., Alnabulsi, A., and Al-Ali, A.R. (2025), "An IoT-Based Mobile Air Pollution Monitoring System", *Lecture Notes in Electrical Engineering*, vol. 1228, pp. 221–233, 326089, doi: https://doi.org/10.1007/978-981-97-4784-9_16
57. Malokhvi E. (2025), "Adaptive filtering and dynamic computation offloading for resilient task execution in IIoT", *Control, Navigation and Communication Systems*, no. 3, pp. 122–127, doi: <https://doi.org/10.26906/SUNZ.2025.3.122>
58. Foss, S., Turlikov, A., and Grankin, M. (2017), "Spatial random multiple access with multiple departure", *IEEE International Symposium on Information Theory - Proceedings*, pp. 2728–2731, 8007025, doi: <https://doi.org/10.1109/ISIT.2017.8007025>
59. Moon, S., Lee, H.-S., and Lee, J.-W. (2018), "SARA: Sparse code multiple access-applied random access for IoT devices", *IEEE Internet of Things Journal*, vol. 5(4), pp. 3160–3174, doi: <https://doi.org/10.1109/JIOT.2018.2835828>
60. Zhu, W., Tao, M., Yuan, X., and Guan, Y. (2023), "Message Passing-Based Joint User Activity Detection and Channel Estimation for Temporally-Correlated Massive Access", *IEEE Transactions on Communications*, vol. 71(6), pp. 3576–3591, doi: <https://doi.org/10.1109/TCOMM.2023.3261382>
61. Rezanov, B., and Kuchuk, H. (2023), "Model of elemental data flow distribution in the Internet of Things supporting Fog platform", *Innovative Technologies and Scientific Solutions for Industries*, no. 3(25), pp. 88–97, doi: <https://doi.org/10.30837/ITSSI.2023.25.088>
62. Gavrylenko, S., Hornostal, O. and Chelak, V. (2022), "Research of Methods of Identifying the Computer Systems State based on Bagging Classifiers", *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)*, pp. 1–6, doi: <https://doi.org/10.1109/KhPIWeek57572.2022.9916439>
63. Kuchuk, G., Nechausov, S. and Kharchenko, V. (2015), "Two-stage optimization of resource allocation for hybrid cloud data store", *International Conference on Information and Digital Technologies, Zilina*, pp. 266–271, DOI: <http://dx.doi.org/10.1109/DT.2015.7222982>
64. Harwahyu, R., Cheng, R.-G., Liu, D.-H., and Sari, R.F (2021), "Fair Configuration Scheme for Random Access in NB-IoT with Multiple Coverage Enhancement Levels", *IEEE Transactions on Mobile Computing*, vol. 20(4), pp. 1408–1419, 8943270, doi: <https://doi.org/10.1109/TMC.2019.2962422>
65. Yu, J., Yu, G., and Chen, Z. (2024), "RALlo: Region Attention-based Edge Resource Allocation in Mobile Internet of Things", *Proc. IEEE Global Comm. Conf. Globecom*, pp. 3413–3418, doi: <https://doi.org/10.1109/GLOBECOM52923.2024.10901347>
66. Liu, J., Wei, X., and Fan, J. (2019), "Tolerable Data Transmission of Mobile Edge Computing under Internet of Things", *IEEE Access*, vol. 7, pp. 71859–71871, 8728032, doi: <https://doi.org/10.1109/ACCESS.2019.2920442>
67. Pardalos, P.M., Steponavičė, I., and Žilinskas, A. (2012), "Pareto set approximation by the method of adjustable weights and successive lexicographic goal programming", *Optimization Letters*, vol. 6(4), pp. 665–678, doi: <https://doi.org/10.1007/s11590-011-0291-5>
68. Śliwiński, T. (2024), "Efficient Approximation Methods for Lexicographic Max-Min Optimization", *Journal of Telecommunications and Information Technology*, no. 1(2024), pp. 46–53, doi: <https://doi.org/10.26636/jtit.2024.1.1421>
69. Zhang, J., Xu, M., and Wang, L. (2025), "Research on Link Selection and Allocation for IoT Localization Systems Based on an Improved Ant Colony Algorithm", *Lecture Notes in Networks and Systems*, 1351 LNNS, pp. 140–150, doi: https://doi.org/10.1007/978-3-031-88287-6_13
70. Zhao, H.-Y., Wang, J.-C., Guan, X., Wang, Z.-H., He, Y.-H., and Xie, H.-L. (2020), "Ant Colony System for Energy Consumption Optimization in Mobile IoT Networks", *Journal of Circuits Systems and Computers*, vol. 29(9), 2050150, doi: <https://doi.org/10.1142/S0218126620501509>

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Малохвій Едуард Едуардович – аспірант, кафедра комп'ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;

Eduard Malokhviy – PhD Student of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;

e-mail: malokhviy.ee@gmail.com; ORCID Author ID: <http://orcid.org/0009-0008-0311-6400>;

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=59179728600>.

Кучук Георгій Анатолійович – доктор технічних наук, професор, професор кафедри комп'ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;

Heorhii Kuchuk – Doctor of Technical Sciences, Professor, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;

e-mail: kuchuk56@ukr.net; ORCID Author ID: <http://orcid.org/0000-0002-2862-438X>;

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=57057781300>.

Коробко Андрій Іванович – доктор технічних наук, професор, провідний науковий співробітник, Харківська філія Українського науково-дослідного інституту прогнозування та випробування техніки і технологій для сільськогосподарського виробництва імені Леоніда Погорілого, Харків, Україна;

Andrii Korobko – Doctor of Technical Sciences, Professor, Leading Researcher, Kharkiv Branch of Leonid Pohorilyu Ukrainian Research Institute for Forecasting and Testing of Agricultural Machinery and Technologies, Kharkiv, Ukraine;

e-mail: ak82andrey@gmail.com; ORCID Author ID: <https://orcid.org/0000-0002-6618-7790>;

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=57215834411>.

Жовнір Наталія Миколаївна – кандидат економічних наук, доцент, доцент кафедри економіки, підприємництва та маркетингу, Національний університет “Полтавська політехніка імені Юрія Кондратюка”, Полтава, Україна;

Nataliia Zhovnir – Candidate of Economic Sciences, Associate Professor, Associate Professor of the Department of Economics, Entrepreneurship and Marketing, National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine;

e-mail: name676@ukr.net; ORCID Author ID: <https://orcid.org/0009-0003-1437-568X>.

Орищенко Сергій Вікторович – кандидат технічних наук, доцент, доцент кафедри тракторів і автомобілів, Національний університет біоресурсів і природокористування України, Київ, Україна;

Serhii Oryshchenko – Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Tractors and Automobiles, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine;

e-mail: oryshchenko.sv@nubip.edu.ua; ORCID Author ID: <https://orcid.org/0000-0002-5359-5285>;

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=57916824300>.

Багатокритеріальний фреймворк для обробки і вивантаження даних на кінцевих пристроях у промисловому IoT

Е. Е. Малохвій, Г. А. Кучук, А. І. Коробко, Н. М. Жовнір, С. В. Орищенко

Анотація. У статті розроблено інтегровану теоретико-математичну структуру для обробки інформації на ресурсно-обмежених кінцевих пристроях Промислового Інтернету речей (IIoT), що функціонують у межах архітектур cloud-fog-edge. **Актуальність дослідження** зумовлена наявністю гетерогенних нестационарних потоків подій, пряма передача яких на вищі рівні обчислювальної ієрархії часто є неможливою через дефіцит пропускної здатності, жорсткі вимоги до затримки, а також енергетичні й обчислювальні обмеження кінцевих пристроїв. У зв'язку з цим кінцевий пристрій повинен виконувати попередню обробку даних, ініційовану процесом сенсорного спостереження, керувати чергами зі скінченими буферами та регулювати вихідний трафік із одночасним збереженням інформативності, необхідної для моніторингу, керування й аналітики. Запропонована формалізація розглядає кінцевий пристрій як активний вузол прийняття рішень, що впливає на динаміку системи шляхом керування локальними перетвореннями та рішеннями щодо вивантаження обчислень за умов змінного в часі ресурсного стану. Модель потоків, чутлива до класів і пріоритетів, відображає неоднорідність критичності та вимог до обслуговування, тоді як динаміка черг зі скінченими буферами описує затримки й втрати за пакетних надходжень і обмеженої пропускної здатності обслуговування. Стан пристрою задається вектором ресурсів, що відображає доступну обчислювальну потужність CPU, зайнятість пам'яті та буфера, якість каналу і швидкість передавання, а також енергетичні обмеження, що дає змогу формулювати залежні від стану умови допустимості локальних обчислень і комунікацій. На рівні операторів систематизовано ланцюг обробки на кінцевому пристрої, який охоплює попередню обробку, оцінювання інформативності, адаптивну фільтрацію, часову та семантичну агрегацію, кероване стиснення й формування компактних ознакових подань. Такий ланцюг формуваний структуровані семантично анотовані пакети, придатні для полегшеного локального прийняття рішень і вибіркового передавання на fog- або cloud-рівні. Запропоновано багатокритеріальну структуру ефективності, що одночасно враховує затримку, втрати пакетів, енергоспоживання, комунікаційне навантаження та збереження інформативності, забезпечуючи тим самим можливість парето-орієнтованого синтезу допустимих адаптивних політик. **Метою дослідження** є встановлення узгоджених змінних керування, системи обмежень, а також умов стійкості й здійсненності, що пов'язують поведінку черг із ресурсними обмеженнями та формують аналітично простежувану основу для подальшої побудови методу, налаштування параметрів і сценарної валідації в реалістичних промислових середовищах. На відміну від суто емпіричного порівняльного аналізу, **внесок роботи** має свідомо аналітичний характер: він інтегрує фрагментовані моделі локального скорочення даних і вивантаження обчислень, а також задає явні операторні визначення для відтворюваного аналізу.

Ключові слова: промисловий Інтернет речей; cloud-fog-edge обчислення; адаптивна фільтрація; агрегація подій; кероване стиснення; модель черг; моделювання ресурсного стану; вивантаження обчислень; багатокритеріальна оптимізація.