UDC 004.7:004.8 DOI: 10.18372/2073-4751.83.20544

Alkema V.V., phd student
orcid.org/0009-0000-0009-8237,
e-mail: vitalii.alkema@gmail.com,

Huzii M.M., Candidate of Technical Sciences,
orcid.org/0000-0003-4807-8862,
e-mail: nn05@ukr.net,

Zozulya Y.A.,
orcid.org/0009-0004-6377-0843,
e-mail: Uzozul@gmail.com

EVENT-DRIVEN EDGE PROCESSING MODEL FOR INDUSTRIAL IoT SYSTEMS

State University "Kyiv Aviation Institute"

Introduction

Efficient management of critical infrastructure assets is essential for ensuring operational continuity, minimizing downtime, and maintaining safety across energy, transport, manufacturing, and other industrial sectors. Even short-term deviations in environmental, technological, or operational parameters may lead to substantial financial losses, equipment failures, or increased risks to personnel and consumers. Traditional IoT-based monitoring systems provide real-time telemetry but remain heavily dependent on cloud connectivity. This dependency introduces several limitations: latency, unstable communication channels, limited on-site autonomy, and insufficient reaction speed in time-critical scenarios.

Recent studies show that shifting data processing to the edge layer significantly improves system responsiveness, resilience, and adaptability. Edge nodes are capable of performing real-time filtering, anomaly detection, event classification, and local decision-making directly at the point where industrial data is generated. However, existing IIoT platforms for monitoring critical infrastructure still predominantly rely on static rule-based logic or cloud-centric processing. As a result, they lack context awareness, dynamic event prioritization, and the ability to operate autonomously during connectivity disruptions.

These challenges highlight the need for an intelligent, event-driven edge computing architecture capable of interpreting contextual information, classifying industrial events by their criticality, and executing on-site decisions in real time. Such an approach enhances the resilience of critical infrastructure, reduces operational delays, and forms a more accurate, robust, and autonomous monitoring ecosystem for the industrial Internet of Things.

Problem Statement

The aim of this research is to develop an **event-driven data processing logic** within an edge-oriented architecture designed for intelligent cold chain logistics management. To achieve this aim, the following objectives are defined:

1. **Analyze the operational characteristics of cold chain logistics** and identify the types of critical events that influence product quality and safety.
2. **Investigate modern IoT and edge computing architectural approaches**, revealing their limitations in terms of latency, autonomy, scalability, and adaptability under real operating conditions.
3. **Develop a model of event processing logic** that provides:

– contextual interpretation of parameter deviations;

- dynamic classification of events based on their severity;
- local decision-making at the edge node;
- minimized dependency on cloud service stability.

Event Processing Logic Model for Intelligent Cold Chain Management

The event processing logic operates directly on the edge node, where sensor measurements are received continuously. To ensure structured interpretation, each event is represented as a four-component tuple shown in formula (1):

$$E = \langle t, s, v, C \rangle (1)$$

Where:

t— timestamp of the measurement;

s— sensor or source identifier;

v— observed value (temperature, humidity, vibration, door state, etc.);

C— context vector containing operational metadata relevant to the event.

The context vector Cprovides additional semantics: current operating mode (loading, steady-state, defrost), recent measurement history, refrigeration cycle state, ambient temperature, and operational flags such as door status or scheduled activity windows. This allows the system to interpret the same numerical deviation differently depending on operational circumstances.

To quantify how strongly an event deviates from expected behaviour, the system computes a deviation descriptor, defined in formula (2):

$$D(E) = \langle \delta_{mag}, \delta_{dur}, \delta_{trend}, \delta_{corr} \rangle (2)$$

Where:

δ_{mag}— magnitude of the deviation relative to the contextual baseline;

δ_{dur}— duration for which the deviation persists;

δ_{trend}— direction and rate of change (increasing, decreasing, oscillatory);

δ_{corr}— correlation with other parameters or events (e.g., compressor off while temperature rising).

This descriptor enables the model to detect both abrupt anomalies and emerging patterns such as slow temperature drift or oscillatory instability.

Based on the deviation descriptor and the context vector, the severity level of the event is determined according to formula (3):

$$L = f_{class}(D(E), C) (3)$$

Where:

f_{class}— adaptive classification function;

L— resulting severity level (informational, warning, critical, or emergency).

Severity is not fixed but adapts over time according to rolling statistics, operator feedback and cloud-provided configuration updates.

After the severity level is determined, the edge node selects a suitable local reaction. This decision process is expressed in formula (4):

$$A = R(L, event\_type, C) (4)$$

Where:

R— decision rule mapping a severity level and event type to an action;

A— resulting action executed locally on the edge node;

event_type — identifier of the anomaly (temperature deviation, compressor anomaly, door event, route deviation, etc.).

Possible actions include recording detailed logs, generating alerts, triggering local alarms, applying safety constraints, or synchronizing results with cloud services once connectivity is restored.

To minimize dependency on cloud stability, the system employs local buffering, deferred synchronization, and compact configuration updates. Long-term analytics, parameter tuning, adaptive threshold recalculation and rule set updates are performed in the cloud and transmitted back to the edge node as lightweight profiles, ensuring continuous operation regardless of connectivity.

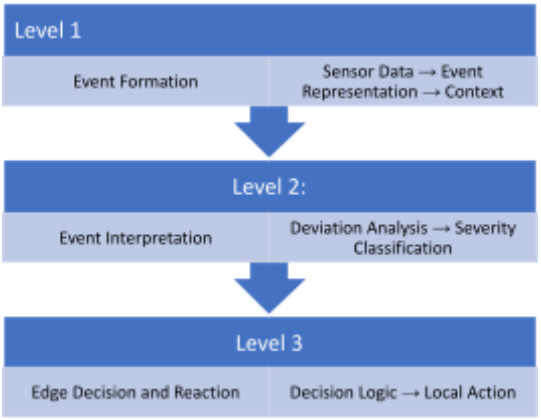

Figure . Hierarchical organization of the event processing logic.

The structured representation in Fig. X illustrates how the proposed event processing logic is organized into three hierarchical levels, showing the sequential transformation of raw sensor inputs into enriched events, analytical interpretations, and finally edge-level decisions. This layered view clarifies the internal workflow of the model and highlights the functional role of each stage within the overall event processing pipeline.

In summary, formulas (1)–(4) collectively define the proposed event processing logic model, which formalizes how raw sensor readings are transformed into contextualized events, how deviations are quantified, how severity levels are dynamically determined, and how appropriate edge-level actions are selected. This formal structure constitutes the core of the model and distinguishes it from conventional threshold-based approaches.

Illustrative Scenario and Modelling Results for Cold Chain Event Processing

This section provides a concise scenario demonstrating how the proposed event processing logic operates under realistic cold-chain transportation conditions. The example illustrates the transformation of raw sensor readings into contextualized events, deviation descriptors, severity levels, and ultimately edge-level actions.

Modelling Tools and Methods

The modelling process used a combination of historical cold-chain telemetry and synthetic perturbations. Historical datasets from refrigerated vehicles provided realistic thermal dynamics, noise characteristics, and compressor behavior. Synthetic time-series components were generated to emulate rare or developing failure modes such as progressive compressor underperformance or weak sensor correlation.

All event tuples, deviation descriptors, and severity classifications were computed using Python-based scripts (NumPy, Pandas, SciPy). Temperature trajectories were reproduced using controlled perturbations to baseline data, preserving thermal inertia and ambient influence. Visual inspection and statistical evaluation of the trajectories were performed using Matplotlib and standard signal-processing utilities. This combination ensured realistic representation of cold-chain operational conditions while enabling controlled evaluation of the event processing logic.

Scenario Description

To illustrate the practical application of the proposed model, consider a refrigerated vehicle transporting frozen goods (−18°C setpoint) along a 120 km route. The edge node inside the vehicle continuously collects temperature, compressor activity, ambient conditions, and door status readings.

At time t_1, the system operates within normal limits:
internal temperature −18.3°C, compressor on, door closed, ambient +4°C.

At time t_2, a small upward drift begins. The temperature increases to −17.6°C while the compressor continues cycling normally. A threshold-based system would not classify this as abnormal.

At time t_3, the drift accelerates:
temperature −16.7°C, compressor cycles become irregular, and the correlation between compressor state and temperature response decreases.

At time t_4, the temperature reaches −15.8°C while compressor output becomes insufficient. Despite being within acceptable regulatory limits, the deviation pattern indicates an emergent equipment malfunction.

Using formula (1), each reading is represented as an event
where Cincludes the thermal history, compressor cycle metadata, ambient temperature, and operational flags.

Using formula (2), the deviation descriptor is calculated. Between $t_3$and t_4:

−δ_{mag}increases from 0.9°C to 2.2°C,

−δ_{dur}shows persistence over 17 minutes,

−δ_{trend}doubles its slope,

−δ_{corr}(compressor–temperature correlation) drops from 0.84 to 0.41.

Using formula (3), the severity classifier assigns the levels:
Informational → Warning → Critical.

Using formula (4), the edge decision logic selects the appropriate reactions:

−local alarm,

−transition to high-power compressor mode,

−buffering diagnostic data locally,

−notifying the driver through the cabin interface.

Even in the absence of cloud connectivity, the edge node autonomously mitigates the developing failure and prevents spoilage.

Modelling Results

To make the behaviour of the proposed event processing logic more explicit, three representative temperature trajectories were modelled and analysed. Each trajectory reflects a typical situation in cold chain operations and demonstrates how the model reacts under different conditions. A summary of the key parameters and outcomes is presented in Table X.

Table . Summary of model behaviour for three temperature trajectories

Scenario	Temperature pattern (inside chamber)	Duration of observation	Severity evolution	Edge-level reaction
Stable Control Curve	Oscillations between −18.4°C and −17.6°C around setpoint	20 min	Informational only	Events logged; no alarms; no control actions triggered
Slow Drift Curve	Gradual rise from −18.0°C to −15.8°C	20 min	Informational → Warning → Critical	Local alarm; compressor forced to high-power mode; driver alert
Door-Abuse Curve	Fast jumps from −18.0°C up to −12.0°C with short cycles	10–12 min	Warning → Emergency	High-priority alarm; immediate notification; event buffering for later cloud analysis

Across these three trajectories, the model consistently distinguished between benign fluctuations and genuinely harmful deviations, escalated severity in a context-aware manner, and triggered edge-level actions without requiring continuous cloud connectivity.

References

1. Чмир О. С., Лисенко В. С. Інтернет речей як інструмент підвищення ефективності логістичних процесів. *Сучасні інформаційні технології у сфері безпеки та оборони*. 2021. № 2. С. 45–52.

2. Костенко О. М., Гуменюк С. О. Технології IoT у моніторингу холодового ланцюга постачання. *Вісник Національного транспортного університету*. 2022. № 1. С. 112–119.

3. Яценко В. О., Дишлевий М. І. Бездротові сенсорні мережі в системах збору телеметричної інформації логістичних підприємств. *Наукові праці ОНАХТ*. 2020. № 3. С. 87–95.

4. Simchi-Levi D., Snyder L. V. Digital transformation in supply chain management: the role of real-time data. *MIT Working Paper*. 2022. 28 p.

5. Rejeb A., Rejeb K., Keogh J. G. Internet of Things in supply chain management: a comprehensive review. *International Journal of Information Management*. 2020. Vol. 52. P. 102–117.

6. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013. Vol. 29(7). P. 1645–1660.

7. Satyanarayanan M. The emergence of edge computing. *Computer*. 2017. Vol. 50(1). P. 30–39.

8. Perera C., Zaslavsky A., Christen P., Georgakopoulos D. Context-aware computing for the Internet of Things: A survey. *IEEE Communications Surveys & Tutorials*. 2014. Vol. 16(1). P. 414–454.

9. Verdouw C. N., Wolfert S., Beulens A. J. M. Smart agri-food logistics: real-time monitoring of storage and distribution. *Journal of Food Engineering*. 2016. Vol. 176. P. 34–41.

10. Akkad M., Döllner J. Event-driven architectures for real-time IoT systems. *Procedia Computer Science*. 2021. Vol. 184. P. 208–215.

11. Wolfert S., Ge L., Verdouw C. N. Big data in smart farming and logistics. *Computers and Electronics in Agriculture*. 2017. Vol. 142. P. 137–153.

12. Голубнича Л., Мандзій Б., Миськів С. Інформаційно-аналітичні системи контролю логістичних процесів. *Науковий вісник НЛТУ*. 2020. № 30(4). С. 125–132.

Alkema V.V., Huzii M.M., Zozulya Y.A.

EVENT-DRIVEN EDGE PROCESSING MODEL FOR INDUSTRIAL IoT SYSTEMS

Efficient and reliable cold chain management requires rapid detection and interpretation of temperature deviations, equipment anomalies, and operational disturbances. Traditional IoT-based monitoring architectures rely heavily on cloud connectivity, resulting in latency, reduced autonomy, and limited resilience in time-critical situations. This paper proposes an event-driven processing logic integrated into an edge computing architecture for intelligent cold chain logistics platforms.

The model introduces a structured four-tuple event representation, a deviation descriptor for quantifying abnormal behaviour, an adaptive severity classification function, and a rule-based edge decision mechanism. Formulas (1)–(4) formalize the transformation of raw sensor measurements into contextualized events, analytically evaluated deviations, severity levels, and real-time edge-level actions.

Modelling results using realistic temperature trajectories—stable control, slow thermal drift, and door-abuse patterns—demonstrate the model's ability to distinguish benign fluctuations from harmful deviations, escalate severity dynamically, and operate autonomously during connectivity interruptions. The proposed approach significantly enhances responsiveness, resilience, and operational reliability of cold chain monitoring systems.

Keywords

edge computing, industrial IoT, event-driven architecture, anomaly detection, contextual processing, real-time decision-making, real-time decision, cyber-physical systems

Алькема В.В., Гузій М.М., Зозуля Ю.А.

Подієво-орієнтована модель граничної обробки даних для систем промислового Інтернету речей

Ефективне управління холодовим ланцюгом потребує швидкого виявлення та інтерпретації температурних відхилень, збоїв обладнання та операційних подій. Традиційні IoT-системи моніторингу значною мірою залежать від хмарних сервісів, що призводить до затримок, обмеженої автономності та зниження надійності у критичних ситуаціях. У статті запропоновано модель подієво-орієнтованої логіки обробки даних у межах edge-архітектури для інтелектуального управління холодовим ланцюгом.

Модель включає формалізоване представлення подій у вигляді чотирикомпонентного кортежу, опис відхилення для кількісної оцінки аномальної поведінки, адаптивну функцію класифікації рівнів критичності та механізм прийняття рішень на рівні edge-вузла. Формули (1)–(4) визначають послідовну трансформацію сирих даних сенсорів у контекстуалізовані події, оцінені відхилення, рівні загрози та локальні дії у режимі реального часу.

Моделювання трьох типових температурних сценаріїв — стабільного режиму, повільного дрейфу та інтенсивних коливань через часте відкривання дверей — підтвердило здатність моделі відрізняти нешкідливі флуктуації від небезпечних відхилень, динамічно підвищувати рівень критичності та функціонувати автономно навіть за відсутності хмарного зв'язку. Запропонований підхід підвищує швидкодію, стійкість та загальну надійність систем холодового моніторингу.

Ключові слова: обчислення на периферії; промисловий Інтернет речей; подієво-орієнтована архітектура; виявлення аномалій; контекстуальна обробка; прийняття рішень у реальному часі; рішення в реальному часі; кіберфізичні системи.