ISSN 2224-087X

ЕЛЕКТРОНІКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ELECTRONICS AND INFORMATION TECHNOLOGIES

Збірник наукових праць

Випуск 35

2026

ELECTRONICS AND INFORMATION TECHNOLOGIES

ЕЛЕКТРОНІКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Issue 35

Scientific journal

Published 4 issue per year

Published since 1966

Випуск 35

Збірник наукових праць

Виходить 4 рази на рік

Видається з 1966 р.

Ivan Franko National University of Lviv

Львівський національний університет імені Івана Франка

2026

ЗАСНОВНИК: ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Друкується за ухвалою Вченої Ради
Львівського національного університету
імені Івана Франка
протокол №14/9 від 30.09.2026 р.

Реєстрація суб'єкта у сфері друкованих медіа:
Рішення Національної ради України з питань
телебачення і радіомовлення № 1877 від
30.05.2024 р. Ідентифікатор медіа R30-04912

У 1966–2010 рр. збірник виходив під назвою «**Теоретична електротехніка»**

Збірник «Електроніка та інформаційні технології» містить оригінальні результати досліджень з електронного матеріалознавства, моделювання фізичних явищ, процесів і систем електроніки, обробки сигналів і зображень, інформаційних технологій.

“Electronics and information technologies” journal contains original research results on electronics material science, modelling of physical phenomena, processes and electronic systems, signal and image processing and information technologies.

Редактори

Проф. *І. Карбовник* – головний співредактор
Проф. *І. Бордун* – головний співредактор
Проф. *О. Крупич* – відповідальний редактор
С.н.с. *Я. Шмигельський* – відповідальний секретар

Редакційна колегія

д-р техн. наук, проф. *О. Андрейків*
д-р габіл., проф. *Б. Андрієвський*
д-р фіз.-мат. наук, проф. *І. Болеста*
канд. фіз.-мат. наук, доц. *С. Вельгош*
д-р фіз.-мат. наук, проф. *П. Венгерський*
д-р техн. наук, проф. *Р. Воробель*
д-р фіз.-мат. наук, проф. *Р. Головчак*
д-р техн. наук, д-р габіл., проф. *Ф. Іващишин*
д-р фіз.-мат. наук, проф. *О. Кушнір*
д-р фіз.-мат. наук, проф. *А. Лучечко*
д-р техн. наук, проф. *Л. Муравський*
д-р техн. наук, проф. *М. Назаркевич*
д-р фіз.-мат. наук, проф. *І. Оленич*
д-р фіз.-мат. наук, проф. *Б. Павлик*
д-р техн. наук, доц. *Б. Павлишенко*
д-р техн. наук, проф. *С. Рендзіняк*
д-р фіз.-мат. наук, проф. *Б. Русин*
д-р фіз.-мат. наук, проф. *М. Притула*
д-р габіл., проф. *Ц. Славінські*
канд. фіз.-мат. наук, доц. *Ю. Фургала*
д-р габіл., проф. *Б. Ціж*,
д-р габіл., проф. *Бушта Шахрауі*
д-р фіз.-мат. наук, проф. *Г. Шинкаренко*
канд. фіз.-мат. наук, доц. *Р. Шувар*
д-р фіз.-мат. наук, проф. *І. Яворський*

Editors

Prof. *I. Karbovnyk* – Chief Co-Editor
Prof. *I. Bordun* – Chief Co-Editor
Prof. *O. Krupych* – Managing Editor
Sen. Res. *Ya. Shmygelsky* – Technical Editor

Editorial Board

O. Andreykiv, Dr. Sc., Prof.
B. Andrievsky, Dr. Habil., Prof.
I. Bolesta, Dr. Sc., Prof.
S. Velgosh, PhD, Assoc. Prof.
P. Vengersky, Dr. Sc., Prof.
R. Vorobel, Dr. Sc., Prof.
R. Holovchak, Dr. Sc., Prof.
F. Ivaschyshyn, Dr. Sc., Dr. Habil., Prof.
O. Kushnir, Dr. Sc., Prof.
A. Luchechko, Dr. Sc., Prof.
L. Muravsky, Dr. Sc., Prof.
M. Nazarkevych, Dr. Sc., Prof.
I. Olenych, Dr. Sc., Prof.
B. Pavlyk, Dr. Sc., Prof.
B. Pavlyshenko, Dr. Sc., Assoc. Prof.
S. Rendzinyak, Dr. Sc., Prof.
B. Rusyn, Dr. Sc., Prof.
M. Prytula, Dr. Sc., Prof.
C. Slawinski, Dr. Habil., Prof.
Yu. Furgala, PhD, Assoc. Prof.
B. Tsizh, Dr. Habil., Prof.
Bouchta SAHRAOUI, Dr. Habil., Prof.
G. Shynkarenko, Dr. Sc., Prof.
R. Shuvar, PhD, Assoc. Prof.
I. Yavorskyy, Dr. Sc., Dr. Habil., Prof.

Адреса редакційної колегії:
Львівський національний університет імені Івана Франка, факультет електроніки та інформаційних технологій, вул. Ген. М. Тарнавського, 107, 79017, Львів, Україна
тел. (+38) (093) 864-01-19

Editorial office address:
Ivan Franko National University of L'viv,
Faculty of Electronics and Computer Technologies
107 Tarnavsky St., UA–79017,
Lviv, Ukraine
tel. (+38) (093) 864-01-19

e-mail: elit@lnu.edu.ua
web-сайт: http://publications.lnu.edu.ua/collections/index.php/electronics/index

АДРЕСА РЕДАКЦІЇ, ВИДАВЦЯ І ВИГОТОВЛЮВАЧА:
Львівський національний університет імені Івана Франка
вул. Університетська, 1, 79000 Львів, Україна

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції. Серія ДК № 3059 від 13.12.2007 р.

© Львівський національний університет імені Івана Франка, 2026

CONTENTS

INFORMATION SYSTEMS AND TECHNOLOGIES

Architecture of ai-augmented virtual learning communities based on community maturity and the 4A knowledge co-creation framework. *Ihor Katernyak, Kiril Fedotov, Viktoriya Loboda (12)* 5

Hardware-aware hybrid compression of convolutional neural networks for resource-constrained systems. *Andrian Kozynets & Lidia Demkiv (14)* 17

Compression coefficient in the analysis of noisy texts and images. *Oleksii Kravchuk & Oleh Kushnir (14)* 65

Segmentation of complex structures in point clouds using deep neural networks. *Viktor Vdovychenko & Yuriy Furgala (12)* 45

Modern architectural principles for designing high-performance distributed systems. *Ivan Rovetskii (20)* 57

A review of cloud-based audio processing using directed acyclic signal-processing graphs in microservice architectures. *Oleh Chaplia, Halyna Klym (20)* 77

MODELING OF PROCESSES AND EFFECTS

Analysis and forecasting of river water pollution dynamics using long short-term memory neural networks and multifractal analysis. *Mykhailo Bordun & Yaroslav Sokolovskyy (12)*. 97

Surrogate modeling of elastic-plastic sheet metal bending: a comparative study of surrogate methods and a diagnostic test of physics regularization. *Bohdan Petrovskyi (12)* 109

Distributed anomaly detection system in manufacturing using edge ai and computer vision. *Olha Shymchyshyn, Kostiantyn Pidhornyi, Maryan Shymchyshyn (10)* 121

INTERNET OF THINGS SYSTEMS

Early detection of anomalies in energy consumption using edge AI. *Roman Korostenskyi (10)* 131

Automated access control system for a smart house using Raspberry PI and automatic number-plate recognition. *Marko Tabachyshyn, Danylo Tabachyshyn, Nataliia Kunanets, Andrii Spitzer (20)* 141

Methods for fault-tolerant and cyber-resilient cloud-oriented iot platforms under embedded device constraints. *Volodymyr Voitovych, Roman Lys (14)* 161

ELECTRONIC DEVICES

Integration of graphene-sensor-based electronic nose technology into mobile microcontroller platform. *Igor Olenych, Yaroslav Boyko, Bohdana Kukharska, Roman Yaremyk, Oleh Sinkevych, Mykhaylo Pavlyk & Ivan Karbovnyk (10)* 175

Optimization of an automated Arduino-based system for characterization of discrete optoelectronic devices. *Mykhailo Rusyniak, Yuriy Korchak, Bohdan Kuzhel (20)* 185

Raspberry PI 5-based multichannel measurement system for gas-flow resistance and temperature monitoring in a natural zeolite packed bed. *Grzegorz Walowski, Dorota Anders, Barbara Dybek, Vadym Ptashnyk, Serhiy Syrotyuk, Patryk Radek (14)* 205

MATERIALS FOR ELECTRONIC ENGINEERING

Spectral and electrical characteristics of Mn- AND Mn-Zr-doped $BaGa_2O_4$ phosphors. *Dmytro Slobodzyan, Markiyan Kushlyk, Vyacheslav Vasyltsiv, Ivanna Medvid, Andriy Luchechko (14)* 219

ЗМІСТ

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Архітектура віртуальних навчальних спільнот, доповнених штучним інтелектом, на основі зрілості спільноти та моделі спільного створення знань 4A. *Ігор Катерняк, Кирил Федотов , Вікторія Лобода (12)* 5

Апаратно-орієнтована гібридна компресія згорткових нейронних мереж для систем з обмеженими ресурсами. *Андріан Козинець, Лідія Демків (14)* 17

Коефіцієнт стиснення в аналізі текстів і зображень із шумами. *Олексій Кравчук, Олег Кушнір (14)* 65

Сегментація складних структур у хмарах точок за допомогою глибинних нейронних мереж. *Віктор Вдовиченко, Юрій Фургала (12)* 45

Сучасні архітектурні принципи проектування високонавантажених розподілених систем. *Іван Ровецький (20)* 57

Огляд методів хмарного опрацювання звуку на основі орієнтованих ациклічних графів оброблення сигналів у мікросервісних архітектурах. *Олег Чапля, Галина Клим. (20)* 57

МОДЕЛЮВАННЯ ПРОЦЕСІВ ТА ЯВИЩ

Аналіз та прогнозування динаміки забруднення річкових вод за допомогою нейронних мереж довгої короткострокової пам'яті та мультифрактального аналізу. *Михайло Бордун, Ярослав Соколовський (12)* 97

Апроксимаційне моделювання пружно-пластичного згинання листового металу: порівняльне дослідження апроксимаційних методів із діагностичною перевіркою фізичної регуляризації. *Богдан Петровський (12)* 109

Розподілена система виявлення аномалій на виробництві з використанням периферійного штучного інтелекту та комп'ютерного зору. *Ольга Шимчишин, Костянтин Підгорний, Мар'ян Шимчишин (10)* 121

СИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ

Раннє виявлення аномалій у споживанні енергії за допомогою периферійного штучного інтелекту. *Роман Коростенський (10)* 131

Автоматизована система контролю доступу для розумного будинку з використанням Raspberry PI та автоматичного розпізнавання номерних знаків. *Марко Табачишин, Данило Табачишин, Наталія Кунанець, Андрій Шпіцер (20)* 141

Методи забезпечення відмовостійкості та кіберстійкості хмарно-орієнтованих платформ інтернету речей в умовах обмежень вбудованих пристроїв. *Володимир Войтович, Роман Лис (14)* 161

ЕЛЕКТРОННІ ПРИСТРОЇ

Інтеграція технології електронного носу на основі графенового сенсора у мобільну мікроконтролерну платформу. *Ігор Оленич, Ярослав Бойко, Богдана Кухарська, Роман Яремик, Олег Сінькевич, Михайло Павлик, Іван Карбовник (10)* 175

Оптимізація автоматизованої системи дослідження дискретних оптоелектронних пристроїв з використанням мікроконтролера Arduino. *Михайло Русиняк, Юрій Корчак*, Богдан Кужель (20)* 185

Багатоканальна вимірювальна система на базі Raspberry PI 5 для дослідження опору газовому потоку та моніторингу температури в шарі природного цеоліту. *Ґжегож Вальовські, Дорота Андерс, Барбара Дибек, Вадим Пташник, Сергій Сиротюк, Патрик Радек (14)* 205

МАТЕРІАЛИ ЕЛЕКТРОННОЇ ТЕХНІКИ

Спектральні та електричні характеристики люмінофорів $BaGa_2O_4$, легованих Mn ТА Mn-Zr. *Дмитро Слободзян, Маркіян Кушлик, В'ячеслав Васильців, Іванна Медвідь, Андрій Лучечко (12)* 219

ISSN 2224-088X (print) • ISSN 2224-0888 (on-line)

Electronics and Information Technologies, 2026, 35, 5–16
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.8:37.018.433

ARCHITECTURE OF AI-AUGMENTED VIRTUAL LEARNING COMMUNITIES BASED ON COMMUNITY MATURITY AND THE 4A KNOWLEDGE CO-CREATION FRAMEWORK

***Ihor Katernyak*[1] **, Kiril Fedotov*[1] *, Viktoriya Loboda*[2]**

Ivan Franko National University of Lviv,
[1] Faculty of Electronics and Computer Technologies,
107 Henerala Tarnavskoho Street, Lviv, 79017, UKRAINE
[2] Faculty of Pedagogical Education,
7 Tuhan-Baranovskoho Street, Lviv, 79005, UKRAINE

Katernyak, I., Fedotov, K., & Loboda, V. (2026). Architecture of AI-Augmented Virtual Learning Communities Based on Community Maturity and the 4A Knowledge Co-Creation Framework. *Electronics and Information Technologies*, 35, 5–16. https://doi.org/10.30970/eli.35.1

ABSTRACT

Background. Virtual learning communities (VLCs) are increasingly used for professional development, project-based learning, and knowledge co-creation. In such environments, the quality of peer assessment and the emotional climate of participation are important for refining project concepts and sustaining engagement. This paper examines the VLC Innovation Spring and presents the pedagogical rationale and software architecture of an AI-augmented facilitation layer designed to support peer assessment and formative feedback.

Materials and Methods. The study draws on data from three annual VLC Innovation Spring cycles with a cumulative sample of 126 participants. The pedagogical design is grounded in a VLC maturity model and the 4A eLearning approach. The platform combines H5P-based interactivity, peer review, and emotional-state monitoring informed by Csikszentmihalyi's flow model. To address variability in the quality of peer reviews, an asynchronous AI facilitation architecture was designed as an external microservice integrated with the VLC platform.

Results and Discussion. The findings confirm the viability of peer learning in the VLC environment. Emotional-state monitoring conducted during one VLC cycle showed that project concept preparation was associated with flow (33.3%) and high arousal (79.4%), whereas peer assessment was associated with a particularly strong sense of control (79.7%). Combined with participants' positive evaluations of the usefulness of the sequence "project concept preparation → peer assessment," these results suggest that the VLC provided an engaging and supportive environment for co-creation. At the same time, substantial variability in the quality of peer reviews emerged as a key limitation.

Conclusion. The proposed asynchronous AI-assisted facilitation architecture extends the VLC with a scalable mechanism for refining project concepts and providing structured formative feedback, without overloading the core platform. The study contributes both pedagogically, by demonstrating the role of emotional-state monitoring in designing a productive co-creation environment, and technically, by proposing an event-driven architecture for AI-supported peer learning in VLCs.

Keywords: virtual learning community, knowledge co-creation, AI facilitation, peer assessment, 4A model, emotional state monitoring.

© 2026 Ihor Katernyak et. al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

Virtual learning communities (VLCs) increasingly function not only as online environments for content delivery and communication, but also as socio-technical systems that support project-based learning, peer interaction, collaborative problem solving, and knowledge co-creation. In such environments, the effectiveness of learning depends not only on pedagogical design but also on the architecture of the digital platform that mediates interaction, feedback, and iterative concept development. This is especially relevant in learning scenarios where participants are supported in developing their higher-order thinking skills [14], including generating project ideas, reviewing peers' work, and refining their concepts through repeated cycles of feedback and reflection.

The broader context for this study is the growing emphasis on knowledge co-creation as a driver of innovation, problem solving, and social learning in digitally mediated environments [1], [2]. Communities of practice and related participatory learning models have long demonstrated that knowledge is constructed through joint activity, legitimate peripheral participation, and the gradual development of shared repertoires [3], [4]. In online settings, these principles are increasingly translated into digital communities where participants collaboratively create, evaluate, and refine ideas, concepts, and artefacts [5], [6], [7]. Such communities are particularly important in entrepreneurship education and innovation-oriented professional development, where the value of learning lies not only in individual acquisition of knowledge but also in the ability to co-create actionable project concepts in interaction with others.

Within project-based VLCs, peer assessment plays a dual role. Pedagogically, it supports reflection, dialogue, reciprocal learning, and a deeper understanding of evaluation criteria. Technically, however, it introduces a substantial system-design challenge: the quality of feedback may vary considerably across participants. Some reviews are detailed, constructive, and well aligned with assessment criteria, while others remain generic, superficial, or weakly connected to the task structure. In large or distributed learning communities, such variation may reduce the perceived fairness and usefulness of peer learning and create a bottleneck in the concept-refinement process. Recent research on online peer feedback and digital peer-review systems confirms that although peer assessment can strengthen self-regulated learning and collaborative reflection, its educational value depends heavily on the quality, specificity, and criterion alignment of feedback, as well as on the design of the supporting digital environment [8], [9]. The present article addresses this challenge in the context of the Virtual Learning Community Innovation Spring, an online environment designed to support participants interested in innovation, entrepreneurship, and sustainable development. Within this VLC, participants develop project concepts, engage in peer assessment, and iteratively refine their ideas through collaborative interaction. The pedagogical design of the community combines two conceptual models. The first is a VLC maturity model consisting of joint venture, mutual engagement, and shared repertoire, rooted in the communities of practice tradition [2–5, 7]. The second is the 4A approach to eLearning, operationalized in this study as Attention, Actualization, Attraction, and Action/Reflection [5]. Together, these models structure the movement from challenge identification and project concept preparation to peer review, reflection, and revision. Earlier work on the Innovation Spring environment and related communities confirmed the importance of social presence, co-creation, and community-based eLearning for sustainable development and collaborative learning [5], [7]. The present study extends that line of work by focusing on a specific bottleneck observed in the mutual-engagement phase: the uneven quality of peer-generated feedback during project concept review. This problem is pedagogically important because peer feedback is expected to help participants refine project logic, clarify objectives, strengthen alignment between challenges and activities, and identify weaknesses in their proposed interventions. Yet when feedback is inconsistent in quality, its formative value decreases, and the burden of interpretation shifts back to learners themselves.

To address this limitation, the paper proposes an asynchronous AI facilitation layer implemented as an external microservice integrated with the VLC platform. The architecture is not intended to replace human reviewers. Rather, it is designed as an AI-assisted facilitation support mechanism that can provide structured semantic analysis of submitted project concepts, generate pre-review recommendations, and scaffold more rubric-oriented peer feedback. In this sense, the study is situated at the intersection of educational technology, learning analytics, and AI-supported assessment. Recent work on large language models (LLMs) in higher education has highlighted their potential to support sustainable feedback practices and formative assessment workflows when embedded in pedagogically grounded and technologically transparent systems [10]. Furthermore, the integration of generative AI into peer assessment introduces new challenges related to academic integrity, algorithmic bias, and ethical data use, requiring strict pseudonymization and transparent fallback protocols [15], [16]. At the same time, studies of collaborative and gamified online learning environments show that emotional engagement and learner experience remain central to the design of effective digital learning ecosystems [11], [12].

The aim of this article is therefore twofold. First, it analyses the pedagogical and emotional dynamics of knowledge co-creation in the VLC Innovation Spring, with particular attention to peer learning and peer assessment as key stages of project concept development. Second, it presents the architecture of an asynchronous human–AI facilitation layer designed to strengthen formative feedback processes in the VLC by reducing variability in peer-review quality and by supporting a more scalable and technically robust model of collaborative concept refinement.

MATERIALS AND METHODS

The study combines two components:

1. an empirical analysis of the operation of a Virtual Learning Community (VLC) supporting project-based learning and peer assessment; and
2. the design of an asynchronous AI facilitation architecture intended to support formative feedback within that VLC.

The primary contribution of the paper is architectural and design-oriented. The empirical component is included to characterize the pedagogical workflow of the VLC, identify the limitations of the existing peer-assessment process, and justify the need for a facilitation layer capable of supporting more consistent formative interaction. The study therefore does not evaluate the causal effectiveness of the AI facilitator as a completed intervention; instead, it presents the architecture as a prototype hybrid support mechanism derived from observed needs in the VLC.

The VLC Innovation Spring serves as the application environment for the proposed facilitation architecture. Its learning workflow is organized into three stages corresponding to the maturity logic of the community of practice (CoP):

Stage I: Joint venture and Concept Submission – the first stage functions as the initialization layer of the collaborative environment. Participants form joint venture teams and articulate their project concepts through structured metadata inputs, defining current needs, proposed solutions, and cost-benefit ratios. This data serves as the foundation for the business model logic. Utilizing benchmarking against existing industry models, participants iteratively refine their value propositions. The system facilitates continuous self-reflection by comparing user-generated models with expert benchmarks, identifying technological and marketing risks before the formal peer review pipeline is triggered.

Stage II: Mutual Engagement (AI Facilitator and Peer Assessment). The second stage introduces a decentralized evaluation mechanism where participants engage in peer assessment. By evaluating three external project concepts, users act as system validators, identifying structural weaknesses and proposing actionable improvements. This asynchronous evaluation process generates a continuous feedback loop. Based on the

system surveys, 98.5% of participants found peer-generated recommendations useful. However, the analysis also revealed a perceived reciprocity imbalance within the peer assessment loop. Participants who invested significant effort in producing high-quality, constructive feedback for others often reported receiving low-quality or superficial evaluations in return. This asymmetry between effort contributed and feedback received led to frustration and reduced perceived fairness of the system, particularly when participants expected similarly substantive input to improve their own project work. While overall feedback utility remained high, this imbalance indicates the need for mechanisms that better align contribution quality with received evaluation quality.

Stage III: Shared Repertoire of new knowledge/practice (Feedback retrieval). The final stage focuses on the practical validation of the generated project solutions. Participants execute pilot testing and target audience interviews to ensure their prototype/minimum viable products (MVPs) align with external requirements. Following this data-driven validation, project parameters – Including timelines, revenue projections, and resource constraints – are finalized. The workflow culminates in a multi-stakeholder presentation platform (Innovation Spring forums and online conferences), where synthesized knowledge is disseminated to external partners and investors, validating the scalability and real-world applicability of the co-created concepts.

In 2022-2025, datasets were collected and organized in the framework with two dimensions: VLC maturity (Joint venture, Mutual engagement and Shared repertoire of new knowledge/practice) and 4A model of eLearning (Attention, Actualization, Attraction and Action/Reflection). The developed knowledge co-creation frame is used for instructional design and evaluation of the learning process (**Table 1**).

Table 1. **eLearning and VLC integrated in one knowledge co-creation frame**

eLearning (4A)	Attention to...	Actualization by...	Attraction through...	Action & Reflection
VLC: Joint Venture	Domain (subject)	VLC goals	Responsibility for team members to accomplish task(s)	Product, project or other result
VLC: Mutual Engagement	Existing experience & practice	Success factors and lessons learned	Professional identity	Presenting own experience. Peer assessment and learning from others
VLC: Shared repertoire	Job to be done	Problems & solutions	Design thinking: from needs to benefits	Networking, knowledge co-creation & dissemination

Implementation of Reward-Driven Engagement Heuristics.

To maximize user engagement and ensure a high completion rate within the VLC, the system architecture integrates reward-driven heuristics using H5P (HTML5 Package) interactive modules. These elements include dynamic quizzes and drag-and-drop assessments that operate as continuous knowledge-validation checkpoints.

Furthermore, gamification is embedded into the core workflow to support students' motivation, engagement, and enjoyment [12] via algorithmic badge allocation. The system triggers specific rewards based on user output quality: for instance, recognizing high-quality concept development and rigorous peer assessment. This design maintains intrinsic motivation, supports a positive competitive environment, and serves as a quantifiable metric for user readiness before interacting with external stakeholders.

AI-Augmented Facilitation Architecture in the VLC.

While the original VLC platform effectively operates as a monolithic web-based system developed in PHP, the integration of Artificial Intelligence requires a decoupled and scalable approach in order to preserve the stability and performance of the core learning environment. To support continuous feedback loops without blocking the main application threads, the proposed AI facilitator is designed as an independent microservice architecture (**Fig. 1**). At the current stage, this architecture should be understood as a prototype facilitation layer intended to augment formative feedback and concept refinement within the VLC, rather than as a fully autonomous or empirically validated assessment system.

As shown in **Fig. 1**, the facilitation workflow consists of three stages: (I) concept submission, (II) asynchronous AI facilitation, and (III) feedback retrieval. In Stage I, when a participant submits a project concept through the LMS, the platform stores the submission metadata in the database and publishes a “concept_submitted” event via the message broker. At the same time, the learner receives an immediate acknowledgement that the concept has been submitted and is being processed, while the AI-related operations continue asynchronously in the background. This design ensures that the main LMS thread remains responsive and that AI processing does not interfere with the normal operation of the VLC platform.

The core of the AI facilitator is built upon a high-performance asynchronous API pipeline. When a participant submits a project concept or performs a peer assessment, the main LMS triggers an event payload via a message broker. The AI microservice processes this payload asynchronously. This automated pipeline enables adaptive feedback generation and scaffolded concept development in near real-time. By utilizing Large Language Models (LLMs) and prompt orchestration, the system performs semantic analysis of the submitted texts, providing structural recommendations before human peer assessment occurs. This architecture makes it possible to provide near real-time support without overloading the main platform and creates the technical basis for adaptive feedback generation and scaffolded concept development.

The model is instructed to evaluate the presence, clarity, and coherence of key concept elements. To ensure full reproducibility of the proposed architecture, the current prototype utilizes the OpenAI GPT-4o model (API version gpt-4o-2024-05-13), configured with a low randomness parameter (temperature = 0.2) to guarantee deterministic and

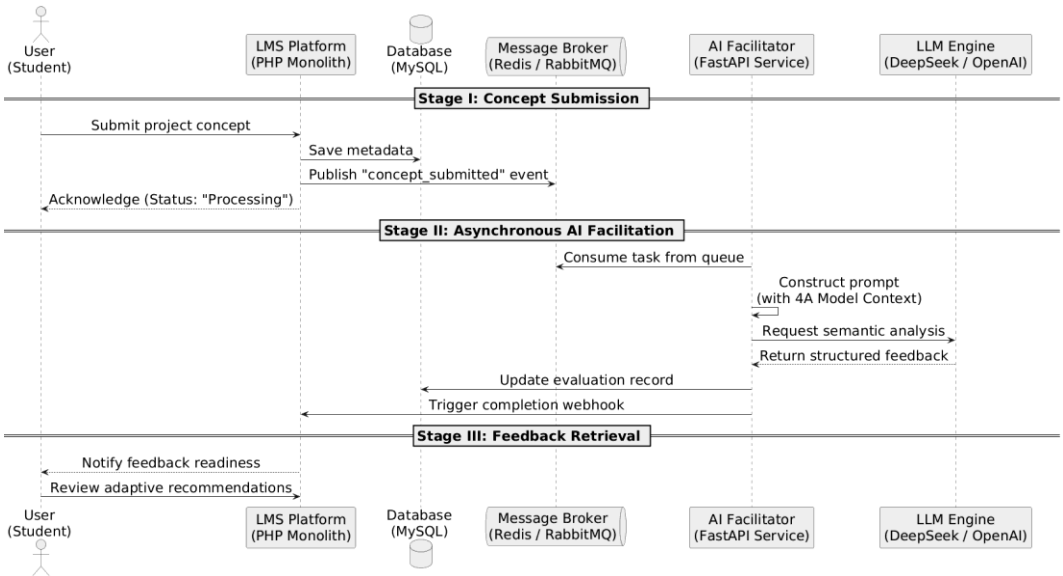


Fig. 1. Architecture of the AI-powered facilitation microservice for VLC evaluation

consistent outputs. The semantic analysis evaluates the concept against four exact rubric parameters on a 0–3 scale: (1) Problem Clarity (accuracy of the challenge and target group definition), (2) Solution Viability (technical and practical feasibility of planned activities), (3) Value Proposition (cost-benefit justification and integration of key partners), and (4) Structural Coherence (internal logic linking identified needs to expected results). To measure automated evaluation alignment, the system calculates the cosine similarity between the vector embeddings of the participant's submission (generated via the text-embedding-3-small model) and the predefined expert benchmarks, applying a baseline similarity threshold of 0.80 for conceptual coherence. The LLM returns structured feedback, which may include comments on missing elements, conceptual ambiguities, weak alignment between objectives and activities, insufficient justification of the challenge, or other issues relevant to the concept paper structure used in the VLC.

Once the model output is returned, the AI facilitator updates the corresponding evaluation record in the database and triggers a completion webhook back to the LMS. In Stage III, the LMS retrieves the processed output and notifies the participant that feedback is available. The learner can then review the AI-generated recommendations in the VLC interface and use them to refine the project concept before peer assessment begins. In this way, the architecture provides near-real-time formative support while keeping the AI functionality technically separated from the monolithic LMS core.

From a pedagogical perspective, the proposed architecture addresses a challenge previously identified in the VLC: the mutual engagement phase may become a bottleneck because the quality, specificity, and criterion alignment of human feedback vary considerably across participants. The AI facilitator is therefore designed as an intelligent scaffolding mechanism that can support more consistent formative interaction without removing the reflective and dialogic role of human reviewers. By generating structured pre-review recommendations and supporting rubric-oriented peer feedback, the system may reduce some of the uncertainty and cognitive effort associated with early-stage concept refinement and peer assessment, particularly for less experienced participants. At the same time, the architecture follows an AI-assisted facilitation principle: AI-generated outputs are advisory and formative rather than authoritative, while final evaluative and interpretive responsibility remains with participants, peer reviewers, and Workshop facilitators.

Because the VLC processes learner-generated texts, the proposed architecture is also designed with privacy and reliability constraints in mind. Only the minimum textual data necessary for semantic analysis should be transferred to the AI microservice, with personal identifiers removed or pseudonymized wherever possible and processing aligned with institutional and legal data-protection requirements. The architecture nevertheless has important limitations: the quality of AI-generated feedback depends on prompt design, rubric quality, and the capabilities of the underlying language model, while semantic similarity between a peer comment and a rubric criterion does not automatically guarantee pedagogical usefulness, contextual appropriateness, or fairness. LLM outputs may also be incomplete, overly generic, or misleading; therefore, fallback scenarios should be provided, such as returning a neutral response or deferring the case to human facilitation when the output is low-confidence or structurally inconsistent. Accordingly, the AI facilitator should be understood as a prototype support mechanism for scaffolding formative feedback and peer review, whose actual educational impact requires separate empirical validation in future research.

RESULTS AND DISCUSSION

For a Virtual Learning Community (VLC) to function as an environment for knowledge co-creation, it must actively support and stimulate the knowledge-sharing behavior of its participants. Through sharing practices, experiences, and project ideas, participants gradually form a cohesive and goal-oriented learning partnership. Knowledge sharing within such a community contributes to value creation, stimulates creativity, and supports

the development of higher-order thinking skills, including innovative idea generation, critical reflection, and the co-construction of knowledge.

At the same time, knowledge-sharing behavior in a VLC does not emerge automatically. Its development requires specific enabling conditions, foremost among them an atmosphere of trust. Trust is essential for participants to present incomplete ideas, expose uncertainties, engage in reciprocal feedback, and accept critique as part of a constructive learning process. For this reason, the development and maintenance of trust within the community are treated as important responsibilities of VLC administrators and coaches.

The key conditions for fostering a trustworthy environment in the VLC include shared values and goals, encouragement of self-presentation, maintenance of professionalism in interaction, and a sense of shared accountability for the quality of participation and feedback. A key pedagogical objective of the virtual learning community (VLC) Innovation Spring was to create a trusting and creative environment that would support idea generation, co-creation, and reciprocal peer learning during the development of project concepts. Emotional-state monitoring was introduced as a reflective instrument for examining whether the VLC atmosphere remained conducive to these processes during two core stages of participation: (I) preparing an individual project concept and (II) conducting peer assessment of project concepts submitted by other participants before final submission to the national forum Innovation Spring.

The monitoring was carried out within one implementation cycle of the VLC Innovation Spring held in the spring. Participants completed the emotional-state item immediately after finishing each of the two tasks on the platform: first, after preparing and submitting their own project concept, and second, after completing the peer-assessment activity. Approximately 50 participants took part in the VLC cycle. The emotional-state question asked participants to indicate which emotional states characterized their experience while performing the task. The response format was based on eight emotional-state categories derived from Csikszentmihalyi's flow model [13]. Participants could select more than one category if several emotional states reflected their experience (**Table 2**).

Table 2. **Summary of the empirical design for the emotional-state analysis**

Dimension	Description
Period	2022–2025
Sample	Three annual VLC Innovation Spring cycles; cumulative N = 126 participants
Instrument	Self-report survey (eight emotional-state categories based on Csikszentmihalyi's flow model)
Indicators	Percentage of specific emotional-state categories reported by participants (e.g., Flow, Arousal, Control)
Purpose	To evaluate emotional engagement and task–skill balance during project concept development (Stage I) and peer assessment (Stage II)

The percentages reported in **Fig. 2** represent the proportion of respondents who selected a given emotional-state category after completing the relevant task. Because multiple responses were allowed, the percentages are not mutually exclusive and do not sum to 100%. The purpose of this monitoring was not to test the statistical effect of an intervention, but to determine whether the VLC design succeeded in maintaining an engaging and supportive atmosphere for concept development and peer review. This interpretation is further supported by participant evaluations of the learning activity: 98.5% of participants reported that the sequence of preparing a project concept and then reviewing the concepts of peers was useful for their learning and project development. Accordingly,

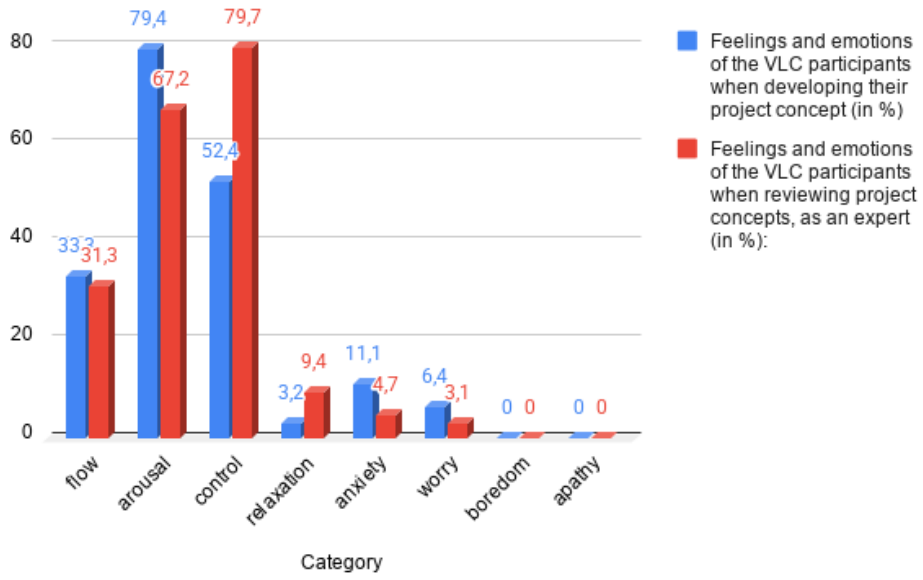


Fig. 2. Comparison of the emotional states of the VLC participants in Stage I (preparing a concept) and Stage II (conducting peer assessment)

the emotional-state results are interpreted in this study as descriptive evidence of the pedagogical viability of the VLC design rather than as the basis for causal claims.

Because the empirical component is intended to characterize the pedagogical workflow and motivate the facilitation layer rather than to test a causal hypothesis, the reported values are descriptive proportions and are not accompanied by inferential statistical testing or confidence intervals; establishing such estimates is left to the future validation study outlined in the Conclusion.

In Stage I of the VLC development process, which focused on project concept preparation, 33.3% of participants reported being in a state of flow. At the same time, a large majority experienced arousal (79.4%), while 52.4% indicated that they felt in control of the process. This emotional profile suggests that concept development was perceived as a highly engaging and stimulating activity. The combination of strong arousal with a substantial sense of control indicates that the task was sufficiently challenging to activate participants while still remaining manageable in relation to their knowledge and skills.

A similar pattern was observed in Stage II, which involved peer assessment of project concepts. To better understand participants' emotional experience during this stage, they were again asked to indicate their feelings and emotions. The results show that the proportion of participants reporting flow remained almost unchanged, accounting for 31.3% in Stage II compared to 33.3% in Stage I. At the same time, the sense of control increased substantially during peer assessment: 79.7% of participants reported that they felt everything was under control. Arousal also remained high, being reported by 67.2% of respondents.

A comparison of emotional states across the two stages shows that flow, arousal, and control dominated in both cases. This finding suggests that the VLC design was able to maintain a productive balance between the level of challenge posed by the assignments and the level of participants' knowledge and skills. However, the emotional configuration differed slightly between the two stages. Stage I was characterized by a higher level of arousal, which may be explained by the personal involvement associated with generating and formulating one's own project concept. In contrast, Stage II was associated with a much stronger sense of control, suggesting that peer assessment enabled participants to engage with the task in a more structured and confident way.

This difference may be pedagogically meaningful. Developing one's own concept requires participants to formulate an idea, articulate its internal logic, and make decisions under conditions of uncertainty, which may explain the higher level of activation during Stage I. Peer assessment, in turn, appears to create conditions in which participants can apply newly acquired knowledge and skills with greater confidence. In this sense, the reviewer role within the VLC does not merely serve an evaluative function; it also reinforces participants' sense of competence and agency by positioning them as active contributors capable of making judgements and providing feedback to others.

Another important observation is the absence of disengaging emotional states. As shown in **Fig. 2**, boredom and apathy were not reported by participants. This indicates that the learning tasks were not experienced as routine, passive, or emotionally detached. On the contrary, the emotional profiles suggest that both concept development and peer assessment sustained participants' attention, involvement, and cognitive-emotional investment throughout the VLC process.

Overall, the emotional responses reported by VLC participants indicate that the learning environment was affectively engaging and that challenge and competence were relatively well aligned across both stages. The persistence of flow, the high level of arousal during concept development, and the particularly strong sense of control during peer assessment all point to the pedagogical value of combining project creation with structured peer review. These findings support the interpretation that the VLC did not function merely as a digital space for submitting tasks, but as a meaningful learning environment that fostered engagement, confidence, and active participation in project-based learning.

At the same time, the findings also reveal an important limitation of the mutual-engagement phase. Although participants generally perceived peer assessment as a manageable and meaningful activity, the quality of reviewer comments remained uneven, ranging from detailed and criterion-related suggestions to highly generic statements with limited formative value. This observation is consistent with recent syntheses of online peer-feedback research, which indicate that the effectiveness of peer review in higher education depends not only on participation rates but also on the structure, specificity, and actionability of comments produced within digital learning environments [8]. Similarly, reviews of online peer-review and assessment systems show that the educational potential of such systems is often constrained by insufficient support for feedback quality control, reviewer calibration, and consistent alignment with evaluation criteria [9].

Taken together, these results suggest that the peer-assessment stage is a high-value point for technological augmentation within the VLC. The proposed asynchronous AI facilitation architecture responds to this need by introducing a support component that can analyze submitted concepts, generate structured formative feedback, and scaffold rubric-oriented review without interrupting the operation of the main platform. Because the architecture is event-driven and external to the VLC core, it also offers a technically scalable model for extending legacy educational systems with AI functionality while preserving platform stability.

In this context, the present study interprets the AI facilitation layer not as an empirically validated substitute for human peer assessment, but as an architectural response to a documented bottleneck in the VLC workflow. Its role is to introduce an additional formative layer that can support interpretation of concepts, surface potential weaknesses in project drafts, and provide participants with more structured entry points for subsequent human peer review. This interpretation is also aligned with emerging work on LLM-supported assessment and feedback, which frames such systems as pedagogically mediated support tools rather than autonomous evaluators [10].

At the same time, the present study does not claim that the proposed architecture has already demonstrated a measurable effect on concept quality, peer-feedback quality, or emotional engagement. Such claims would require a separate empirical evaluation comparing learning outcomes with and without the AI facilitation layer. The contribution of this paper is therefore more modest but also more precise: it identifies a pedagogically

meaningful bottleneck in a functioning VLC and proposes a technically grounded architecture for addressing it.

CONCLUSION

This study examined knowledge co-creation in the Virtual Learning Community Innovation Spring and identified peer assessment as both a pedagogically valuable and structurally vulnerable stage of the learning process. The empirical analysis showed that concept preparation and peer review were associated with productive emotional states, indicating that participants experienced the VLC as a meaningful environment for project-based learning and collaborative interaction. At the same time, the quality of peer-generated feedback varied substantially, creating a bottleneck in the mutual-engagement phase of the community.

In response to this limitation, the article proposed an asynchronous human–AI facilitation architecture designed as an external microservice integrated with the VLC platform. By combining event-driven communication, a FastAPI-based service layer, prompt orchestration, and LLM-based semantic analysis, the proposed architecture extends the VLC with a technically grounded mechanism for supporting concept refinement and peer feedback without overloading the core platform.

The contribution of the article lies in linking empirical evidence about peer-learning dynamics and emotional engagement in a virtual learning community with the design of an AI-supported facilitation layer aimed at strengthening formative interaction. The study does not claim a separately validated causal effect of the AI facilitator on learning outcomes. Rather, it demonstrates how empirical analysis of a VLC can inform the design of a scalable, modular, and pedagogically oriented AI extension for knowledge co-creation environments in higher education and professional development.

Future research should focus on evaluating the effectiveness of the AI facilitation layer after deployment, including its influence on the quality of project concepts, the specificity and usefulness of feedback, revision behavior, and participant experience. Such evaluation will be essential for determining whether the proposed architecture can serve not only as a technically elegant solution, but also as an educationally effective component of hybrid human–AI learning ecosystems.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no conflicts of interest relevant to this study.

AUTHOR CONTRIBUTIONS

Conceptualization, [I.K.]; methodology, [I.K., V.L.]; software, [K.F.]; validation, [I.K., V.L., K.F.]; formal analysis, [I.K., K.F.]; investigation, [V.L.]; resources, [I.K.]; data curation, [I.K., V.L.]; writing – original draft preparation, [I.K., V.L., K.F.]; writing – review and editing, [I.K., V.L.]; visualization, [K.F.]; supervision, [I.K.]; project administration, [I.K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Kreiling, L., & Paunov, C. (2021). *Knowledge co-creation in the 21st century: A cross-country experience-based policy report*. OECD Science, Technology and

Industry Policy Papers, No. 115. OECD Publishing, Paris. https://doi.org/10.1787/c067606f-en
[2] Wenger-Trayner, E., & Wenger-Trayner, B. (2020). *Learning to make a difference: Value creation in social learning spaces*. Cambridge University Press. https://doi.org/10.1017/9781108677431
[3] Brown, M., & Peck, C. (2018). Expanding the landscape: Developing knowledgeability through communities of practice. *International Journal for Academic Development, 23*(3), 232–243. https://doi.org/10.1080/1360144X.2018.1473252
[4] Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press. https://doi.org/10.1017/CBO9780511815355
[5] Katernyak, I., Loboda, V., & Kulya, M. (2018). eLearning within the Community of Practice for sustainable development. *Higher Education, Skills and Work-Based Learning, 8*(3), 312–322. https://doi.org/10.1108/HESWBL-03-2018-0030
[6] Yang, S. J. H., Chen, I. Y. L., Kinshuk, & Chen, N. S. (2007). Enhancing the quality of e-learning in virtual learning communities by finding quality learning content and trustworthy collaborators. *Educational Technology & Society, 10*(2), 84–95. https://www.jstor.org/stable/jeductechsoci.10.2.84
[7] Katernyak, I., Sheremet, M., & Loboda, V. (2009). eLearning: From social presence to co-creation in virtual education community. *Interactive Technology and Smart Education, 6*(4), 215–222. https://doi.org/10.1108/17415650911005375
[8] Taghizadeh Kerman, N., Banihashem, S. K., Karami, M., Er, E., van Ginkel, S., & Noroozi, O. (2024). Online peer feedback in higher education: A synthesis of the literature. *Education and Information Technologies, 29*(1), 763–813. https://doi.org/10.1007/s10639-023-12273-8
[9] Babik, D., Gehringer, E., Kidd, J., Sunday, K., Tinapple, D., & Gilbert, S. (2024). A systematic review of educational online peer-review and assessment systems: Charting the landscape. *Educational Technology Research and Development, 72*, 1653–1689. https://doi.org/10.1007/s11423-024-10349-x
[10] Agostini, D., & Picasso, F. (2024). Large language models for sustainable assessment and feedback in higher education: Towards a pedagogical and technological framework. *Interaction Design and Architecture(s), 18*(1), 75–92. https://doi.org/10.3233/IA-240033
[11] Perez-Aranda, J., Medina-Claros, S., & Urrestarazu-Capellán, R. (2024). Effects of a collaborative and gamified online learning methodology on class and test emotions. *Education and Information Technologies, 29*, 1823–1855. https://doi.org/10.1007/s10639-023-11879-2
[12] Acosta-Medina, J., Torres-Barreto, M., & Cárdenas-Parga, A. (2021). Students' preference for the use of gamification in virtual learning environments. *Australasian J. of Educ. Technology, 37*(1), 145–158. https://doi.org/10.14742/ajet.6512
[13] Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. Harper & Row.
[14] Meng, Q., Jia, J., & Zhang, Z. (2020). A framework of smart pedagogy based on the facilitating of high order thinking skills. *Interactive Technology and Smart Education, 17*(3), 251–266. https://doi.org/10.1108/ITSE-11-2019-0076
[15] Perkins, M., Roe, J., Vu, B. H., Postma, D., Hickerson, D., McGaughran, J., & Khuat, H. Q. (2024). Simple techniques to bypass GenAI text detectors: Implications for inclusive education. *International Journal of Educational Technology in Higher Education, 21*, 53. https://doi.org/10.1186/s41239-024-00487-w
[16] Chan, C. K. Y. (2023). A comprehensive AI policy education framework for university teaching and learning. *International Journal of Educational Technology in Higher Education, 20*, 38. https://doi.org/10.1186/s41239-023-00408-3

АРХІТЕКТУРА ВІРТУАЛЬНИХ НАВЧАЛЬНИХ СПІЛЬНОТ, ДОПОВНЕНИХ ШТУЧНИМ ІНТЕЛЕКТОМ, НА ОСНОВІ ЗРІЛОСТІ СПІЛЬНОТИ ТА МОДЕЛІ СПІЛЬНОГО СТВОРЕННЯ ЗНАНЬ 4A

Ігор Катерняк[1], Кирил Федотов[1] , Вікторія Лобода[2],

Львівський національний університет імені Івана Франка,
[1] Факультет електроніки та комп'ютерних технологій,
вул. Генерала Тарнавського 107, Львів, 79017, УКРАЇНА
[2] Факультет педагогічної освіти,
вул. Туган-Барановського, 7, Львів, 79005, УКРАЇНА

АНОТАЦІЯ

Актуальність. Віртуальні навчальні спільноти (ВНС, англ. VLC – virtual learning communities) дедалі ширше використовуються для професійного розвитку, проєктно-орієнтованого навчання та співтворення знань. У таких середовищах якість взаємооцінювання та емоційний клімат участі є важливими для вдосконалення проєктних концепцій і підтримання залученості. У статті досліджено ВНС Innovation Spring та представлено педагогічне обґрунтування і програмну архітектуру ШІ-посиленого фасилітаційного шару для підтримки взаємооцінювання та формувального зворотного зв'язку.

Матеріали та методи. Дослідження спирається на дані трьох щорічних циклів ВНС Innovation Spring із сукупною вибіркою 126 учасників. Педагогічний дизайн ґрунтується на моделі зрілості VLC та підході 4A до eLearning. Платформа поєднує H5P-інтерактивність, взаємооцінювання та моніторинг емоційних станів на основі моделі потоку М. Чіксентмігаї. Для подолання варіативності якості рецензій спроєктовано асинхронну ШІ-фасилітаційну архітектуру у вигляді зовнішнього мікросервісу, інтегрованого з ВНС-платформою.

Результати та обговорення. Результати підтверджують життєздатність взаємонавчання у ВНС-середовищі. Моніторинг емоційних станів, проведений в одному з циклів ВНС, показав, що підготовка проєктної концепції асоціювалася зі станом потоку (33,3%) і високим рівнем збудження (79,4%), тоді як взаємооцінювання — з особливо сильним відчуттям контролю (79,7%). У поєднанні з позитивними оцінками учасників щодо корисності послідовності «підготовка концепції → взаємооцінювання» це свідчить про те, що ВНС забезпечувала залучальне та підтримувальне середовище для співтворення. Водночас суттєва варіативність якості рецензій виявилася ключовим обмеженням.

Висновки. Запропонована асинхронна ШІ-підтримувана фасилітаційна архітектура розширює ВНС масштабованим механізмом доопрацювання проєктних концепцій і структурованого формувального зворотного зв'язку без перевантаження основної платформи. Дослідження поєднує педагогічний і технічний внесок, демонструючи роль моніторингу емоційних станів у проєктуванні середовища співтворення та пропонуючи архітектуру, керовану подіями, для ШІ-підтримуваного взаємонавчання у ВНС.

Ключові слова: віртуальна навчальна спільнота, спільне творення знань, ШІ-фасилітація, взаємооцінювання, модель 4A, моніторинг емоційних станів.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
24 May, 2026	06 July, 2026	08 July, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 17–30
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.89:004.93

HARDWARE-AWARE HYBRID COMPRESSION OF CONVOLUTIONAL NEURAL NETWORKS FOR RESOURCE-CONSTRAINED SYSTEMS

Andrian Kozynets* & *Lidia Demkiv*
System Design Department
Ivan Franko National University of Lviv
50, Drahomanova Street, Lviv, 79005, UKRAINE

Kozynets, A., Demkiv, L., (2026). Hardware-Aware Hybrid Compression of Convolutional Neural Networks for Resource-Constrained Systems. *Electronics and Information Technologies*, 35, 17–30. https://doi.org/10.30970/eli.35.2

ABSTRACT

Background. Deep neural networks face critical computational barriers for deployment on resource-constrained edge devices. While individual compression methods exist, their isolated application causes severe accuracy degradation. The synergistic combination of pruning, quantization, and knowledge distillation presents a solution, yet their interaction and hardware-specific efficiency remain insufficiently explored.

Materials and Methods. This study investigates hybrid compression pipelines applied to linear (VGG) and residual (ResNet) network architectures. We evaluated a Two-Stage Distillation algorithm featuring a preliminary classifier stabilization phase to eliminate gradient shock, combined with structural filter removal and static integer quantization. A cross-platform analysis was conducted on central and graphics processing units, and Pareto efficiency frontiers were constructed to determine the optimal trade-offs.

Results and Discussion. The preliminary stabilization phase successfully mitigated gradient shock, consistently improving student accuracy by up to 2.6%. While isolated structural pruning significantly degraded performance, distillation healing via teacher-generated soft targets enabled a pruned network with nearly 30% of weights removed to outperform its full-sized baseline. On CPUs, the hybrid pipeline reduced memory footprint by 11 times and accelerated inference by 5 times. Conversely, integer quantization demonstrated critical inefficiency on GPUs due to memory latency; for such accelerators, half-precision format was Pareto-optimal for massive linear architectures, whereas standard precision guided by two-stage distillation provided the best efficiency for residual models.

Conclusion. Integrating Two-Stage Distillation with structural pruning and quantization fundamentally outperforms isolated compression techniques. The optimal strategy is strictly hardware- and architecture-dependent: integer-based hybrid pipelines are essential for energy-efficient edge CPUs, while for GPU accelerators, the choice between half-precision and standard precision depends on the network's topological characteristics.

Keywords: Artificial intelligence, Deep learning, Neural network optimization, Resource-constrained systems, Hardware acceleration.

INTRODUCTION

The rapid evolution of artificial intelligence, particularly in the field of computer vision, has been largely driven by the increasing depth and complexity of convolutional neural networks (CNNs). Modern retrospective analyses of deep learning architectures demonstrate that foundational topologies, such as the ResNet and VGG families, achieve unprecedented benchmarks in image classification and object detection by utilizing

© 2026 Andrian Kozynets & Lidia Demkiv. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

massive parametric spaces to extract highly complex, hierarchical feature representations [1]. However, this architectural scaling comes at a severe computational cost. Modern CNNs require gigabytes of memory for weight storage and billions of floating-point operations (FLOPs) per inference. Consequently, the deployment of state-of-the-art CNNs on edge devices – such as mobile platforms, autonomous drones, and Internet of Things (IoT) sensors – is strictly hindered by limited memory bandwidth, restricted thermal envelopes, and stringent energy budgets [2].

To bridge the substantial gap between high-performance deep learning models and the physical hardware constraints of peripheral devices, the scientific community has intensely focused on model compression techniques. Among these, Knowledge Distillation (KD) has emerged as a fundamental and highly effective approach. KD involves training a compact "student" model to replicate the functional behavior of a cumbersome "teacher" model by learning from the teacher's temperature-scaled probabilities (soft targets), which contain crucial dark knowledge regarding inter-class similarities [3]. While KD provides an elegant theoretical framework for information transfer, its isolated application often fails to achieve the extreme compression ratios required for real-time edge inference. Furthermore, recent statistical evaluations of distillation mechanisms indicate that applying KD directly to new, uninitialized domains frequently induces a "gradient shock," where chaotic initial gradients irreversibly disrupt the spatial feature hierarchies of the student model, significantly limiting knowledge absorption [4].

To achieve aggressive model minimization, structural pruning and quantization are actively employed. Structural pruning aims to reduce the model's physical width and computational complexity by completely removing redundant structural components, such as entire convolutional filters or feature channels. Unlike unstructured pruning, structural reduction directly translates to hardware acceleration without the need for specialized sparse matrix libraries [5]. Simultaneously, model quantization converts high-precision 32-bit floating-point (FP32) weights and activations into low-bitwidth integer formats, such as 8-bit integers (INT8) [6]. This allows models to leverage rapid vector instructions on modern Central Processing Units (CPUs), drastically reducing memory footprint and accelerating inference speed.

Despite the individual promise of these techniques, their practical implementation in hybrid pipelines faces profound algorithmic and hardware-related challenges. The simultaneous or sequential application of structural pruning and INT8 quantization radically alters the geometry of the parameter space. Standard fine-tuning using the conventional cross-entropy loss function is often incapable of navigating this severely restricted space, leading to a catastrophic and non-linear degradation in classification accuracy [7]. Furthermore, the efficiency of these compression pipelines is strictly hardware-dependent. A compression configuration optimized for a cloud-based Graphics Processing Unit (GPU) – which favors floating-point operations and massive parallelism – might result in severe memory bottlenecks when deployed on an edge CPU, and vice versa [2].

Therefore, there is an objective need for a formalized, hardware-aware optimization framework that synergizes these distinct compression paradigms. This study investigates the integration of a Two-Stage Distillation algorithm with hybrid compression pipelines to mitigate accuracy degradation. In particular, the phenomenon of “distillation healing” is explored, under the premise that teacher-generated soft targets can act as a precise spatial navigator for the optimizer, thereby recovering the representative capacity of pruned networks within a narrowed search space.

The main objective of this study is to develop, theoretically justify, and experimentally evaluate a hybrid compression framework that integrates Two-Stage Distillation, structural pruning, and static quantization. By conducting an extensive cross-platform analysis on CPU and GPU architectures, this research aims to determine Pareto-efficient configurations that provide an optimal balance between classification accuracy, memory

footprint, and real-time inference speed for resource-constrained artificial intelligence models.

MATERIALS AND METHODS

Neural Network Architectures and Experimental Framework. To ensure a comprehensive and architecture-agnostic evaluation of the proposed hybrid compression pipelines, the experiments were conducted using two fundamentally distinct convolutional neural network topologies. Although lightweight architectures such as MobileNet, EfficientNet, and RepVGG are widely adopted for edge deployment, this study intentionally focuses on the VGG and ResNet families. Recent studies continue to investigate compression and deployment techniques based on ResNet architectures for resource-constrained edge systems, demonstrating their ongoing relevance as benchmark backbones for evaluating optimization methods [8]. Furthermore, recent surveys also emphasize the coexistence of lightweight and conventional CNN architectures in modern edge AI research, where MobileNet, EfficientNet, RepVGG, VGG, and ResNet serve complementary roles depending on application requirements [9]. ResNet remains one of the most frequently used backbone architectures in modern computer vision applications, whereas VGG continues to serve as a widely accepted benchmark model in neural network compression research. Furthermore, these architectures represent two fundamentally different CNN design paradigms: sequential and residual, which enable the evaluation of the proposed compression framework across distinct network topologies. The first family is represented by the VGG architecture (specifically, VGG16 as the teacher and VGG11 as the baseline student). VGG networks are characterized by a deep, linear topology with massive fully connected layers, making them highly susceptible to gradient degradation and memory bottlenecks. The second family includes Residual Networks (ResNet50 as the teacher and ResNet18 as the student). Unlike VGG, ResNet incorporates skip connections (residual blocks) that act as safe gradient highways, providing inherent structural robustness and efficient feature propagation. Recent retrospective analyses confirm that evaluating compression techniques across both linear and residual topologies is mandatory to objectively determine a method's universal applicability [10].

Open-Source Models and Dataset Adaptation. To ensure the reproducibility and transparency of the research, all investigated models (VGG16, VGG11, ResNet50, and ResNet18) are standard, classical architectures. Their implementations and baseline weights were obtained from the publicly available open-source PyTorch repository (*torchvision.models*).

It is important to emphasize that the proposed hybrid compression methodology is fundamentally dataset-agnostic. However, to provide an objective and reproducible experimental baseline, all models were evaluated using the publicly accessible ImageNet100 dataset, a standardized benchmark containing 100 image classes.

Furthermore, the experimental protocol did not involve training the baseline models from scratch. Instead, the pre-trained architectures underwent a preliminary fine-tuning (adaptation) phase on the ImageNet100 dataset. This step was mandatory to align the feature space with the target domain before any destructive compression manipulations (such as structural pruning or INT8 quantization) and the subsequent distillation healing processes were applied. This approach strictly isolates the performance impact of the compression algorithms from the variances of initial training.

Implementation of Compression Algorithms. The core methodology of this study revolves around the synergistic application of destructive compression techniques and knowledge-guided recovery. The following discrete optimization algorithms were implemented:

Structural Filter Pruning. Unlike unstructured pruning, which zeroes out individual weights and requires specialized sparse-matrix hardware to realize speedups, we

implemented structural filter pruning based on the L_1-norm criterion [11]. In this approach, the absolute sum of the weights within each convolutional filter is calculated. Filters with the lowest L_1-norm are considered to contribute the least to the feature extraction process and are physically removed from the network architecture. This reduces the number of output channels, subsequently decreasing the number of multiply-accumulate (MAC) operations. In our experimental setup, a baseline sparsity level of 15% to 28% was targeted for the student models [5].

Static Integer Quantization (INT8). To maximize inference throughput on edge devices lacking powerful floating-point units, post-training static quantization (PTSQ) was utilized [12]. This process maps the 32-bit floating-point (FP32) weights and activations into an 8-bit integer (INT8) space. Because dynamic quantization calculates activation ranges during runtime (adding computational overhead), static quantization was chosen. It requires a preliminary calibration phase, where a representative subset of the validation dataset is passed through the network to compute the statistical distribution (minimum and maximum clipping ranges) of the activations.

Half-Precision Formatting (FP16). To leverage the specific hardware accelerators available on modern Graphics Processing Units (GPUs), a conversion to half-precision floating-point format (FP16) was implemented [13]. This reduces the memory bandwidth requirements by half compared to FP32, accelerating memory-bound operations without the severe precision loss associated with aggressive integer quantization.

Knowledge Distillation as a Healing Mechanism. While the mathematical formulation and initialization stabilization of the Two-Stage Distillation algorithm are outside the scope of this paper, its role in this research acts exclusively as a recovery protocol. Instead of fine-tuning the severely damaged pruned architectures using standard hard-label cross-entropy, the temperature-smoothed soft targets ($T = 4.0$) generated by the robust teacher model are used to guide the student's optimizer. This "distillation healing" provides dense spatial navigation through the narrowed parameter space [3].

Classification of Investigated Compression Pipelines. To systematically isolate the effects of individual and hybrid compression techniques, a rigid matrix of experimental pipelines was established. **Table 1** defines the ten distinct methodologies evaluated in this study.

Table 1. **List of experimental optimization methods**

No	Method name
00	Baseline (Teacher)
01	Quantization Static (INT8)
02	Pruning Structured(15%)
03	Self-Distillation
04	Distillation + Quantization
05	Baseline Student
06	Two-Stage Distillation (Student)
07	Distillation + FP16
08	Distillation + Prune
09	Distillation + Prune + Quantization

Baselines (Methods 00 & 05). Method 00 (Teacher) serves as the theoretical upper bound for accuracy and the source of "dark knowledge" (ResNet50 or VGG16). Method 05

(Baseline Student) represents the lightweight architecture trained from scratch using standard cross-entropy, illustrating the architectural capacity gap.

Isolated Classical Compression (Methods 01 & 02). These pipelines apply compression without distillation guidance. Method 01 involves isolated static INT8 quantization. Method 02 physically removes 15% of convolutional filters using L_1-norm pruning, followed by standard hard-label fine-tuning.

Standard Distillation Indicators (Methods 03 & 04). Method 03 represents the classic self-distillation protocol, forcing a randomly initialized student to match the teacher's KL-divergence immediately. Method 04 follows this with standard quantization. These methods serve to demonstrate the vulnerability of networks to unmitigated gradient shocks.

Advanced Distillation and Healing (Methods 06, 07, & 08). Method 06 is the isolated application of the Two-Stage Distillation protocol. Method 07 converts the distilled student (Method 06) into FP16 format, designed specifically for GPU tensor core acceleration. Method 08 is the core "distillation healing" pipeline, where the student model undergoes aggressive structural pruning and is subsequently fine-tuned using the teacher's soft targets.

Extreme Hybrid Compression (Method 09). This is the most aggressive pipeline, engineered for severely constrained edge environments. It sequentially combines distillation (for maximal feature retention), structural pruning (for topological reduction), and static INT8 quantization (for vector-instruction acceleration).

Hardware Platforms and Cross-Platform Testing Environment. The computational efficiency of any model compression pipeline is fundamentally hardware-dependent. To comprehensively evaluate the proposed hybrid methods, empirical testing was conducted across two distinct hardware environments, representing the two primary deployment scenarios: resource-constrained edge computing and high-performance cloud inference [2].

The first environment, serving as a proxy for edge devices, utilized an AMD Ryzen 5 5600 multi-core Central Processing Unit (CPU) supporting Advanced Vector Extensions (SIMD instructions) and 16 GB of system memory. This platform was explicitly chosen to evaluate the hardware acceleration capabilities of static INT8 quantization (Methods 01, 04, and 09), as integer arithmetic allows the CPU to process vectorized data efficiently, significantly reducing latency [12]. The second environment utilized an NVIDIA GeForce RTX 3060 Ti Graphics Processing Unit (GPU) equipped with Tensor Cores and Compute Unified Device Architecture (CUDA) support. This environment was used to measure baseline models and to test the specific efficacy of half-precision (FP16) formatting (Method 07) [13].

Also, all performance measurements reported in this study were obtained using a batch size of 32.

Evaluation Metrics and Data Processing. To quantitatively assess the effectiveness of the distillation and hybrid compression algorithms, three primary metrics were recorded for each experimental pipeline:

1. Classification Accuracy. Measured as the standard validation accuracy on the testing dataset. This metric serves as the primary indicator of the feature space integrity and the success of the distillation healing mechanism.

2. Memory Footprint and Compression Ratio. The physical storage size of the serialized model weights, measured in Megabytes (MB). The Compression Ratio is calculated as the baseline teacher model size divided by the compressed student model size. This metric is crucial for determining the feasibility of deployment on memory-constrained peripheral devices.

3. Inference Speed (Throughput). Measured in Frames Per Second (FPS). To isolate the computational throughput of the network from data-loading bottlenecks, FPS was calculated based on model forward-pass execution times, allowing for an objective cross-platform speed comparison.

Since no single pipeline simultaneously maximizes accuracy and minimizes latency across all hardware, a Pareto-efficiency analysis was employed. Constructing Pareto

frontiers allows for an objective mathematical comparison of the trade-offs inherent to different compression strategies. All experimental data, including neural network weights, validation accuracies, and inference latency measurements, were collected and processed natively within the Python programming environment using the PyTorch deep learning framework [14].

RESULTS AND DISCUSSION

The "Distillation Healing" Phenomenon: Restoring Capacity After Pruning. A key objective of the experimental phase was to evaluate the capacity of knowledge distillation to preserve feature space integrity during structural pruning. Structural pruning physically removes convolutional filters based on their L_1-norm, radically altering the geometry of the parameter space and permanently reducing the model's capacity [11]. The isolated application of this method, followed by standard fine-tuning using hard categorical labels, typically fails to allow the model to adequately recover its lost representational pathways. This occurs because the narrowed architecture lacks the dense parameterization required to navigate the complex loss landscape from scratch. To evaluate this, we compared isolated pruning against our hybrid approach, wherein the pruned student architecture is trained using the temperature-smoothed "soft targets" generated by the uncompressed teacher model.

The empirical data presented in **Table 2** substantiates the severe impact of unguided compression on the ResNet architecture. The application of standard structural pruning (Method 02) demonstrably degrades the network's accuracy, dropping it to 0.839 relative to the baseline teacher.

Table 2. **The impact of distillation on robustness to structural pruning (ResNet family).**

Method №	Accuracy	Sparsity
01 (Quantization Static (INT8))	0.845	0.020
02 (Pruning Structured(15%))	0.839	0.121
04 (Distillation + Quantization)	0.851	0.020
08 (Distillation + Prune)	0.778	0.287
09 (Distillation + Prune + Quantization)	0.789	0.296

However, the "distillation healing" phenomenon is prominently manifested in Method 08. Following the aggressive removal of approximately 28% of the convolutional weights in the ResNet18 student model, the introduction of teacher-guided soft targets enabled the highly sparse model to achieve an accuracy of 0.778. This results in a highly counterintuitive paradox: the pruned, distilled student model strictly outperforms the fully dense baseline student model trained from scratch (Method 05, which only achieved an accuracy of 0.770).

This outcome provides strong empirical evidence that knowledge distillation does not merely act as an initialization strategy but serves as an ideal spatial navigator. The soft targets contain essential "dark knowledge" about inter-class similarities, which effectively guides the optimizer to find highly efficient signal routing paths within the strictly constrained parameter space of the pruned network [3].

As illustrated in **Table 3**, an analogous "healing" effect is observed for linear topologies, which are traditionally much more vulnerable to structural damage than residual networks due to the lack of gradient bypass connections [15]. Standard pruning applied to the VGG architecture (Method 02) causes a steep collapse in accuracy down to 0.635.

Table 3. **The impact of distillation on robustness to structural pruning (VGG family).**

Method №	Accuracy	Sparsity
01 (Quantization Static (INT8))	0.649	0.017
02 (Pruning Structured(15%))	0.635	0.016
04 (Distillation + Quantization)	0.623	0.017
08 (Distillation + Prune)	0.64	0.498
09 (Distillation + Prune + Quantization)	0.635	0.479

Nevertheless, the application of the hybrid pipeline (Method 08) successfully mitigates this degradation. By fine-tuning the pruned VGG network with distillation gradients, the feature space is partially restored, allowing the network to reach an accuracy of 0.640.

Remarkably, just as observed with the ResNet family, this heavily pruned VGG network outperforms the full-sized baseline VGG11 student (Method 05, accuracy 0.616). This confirms that the distillation healing mechanism is architecture-agnostic and universally beneficial for restoring the capacity of sparse convolutional topologies.

Deep Compression and Memory Footprint Reduction. For the successful deployment of convolutional neural networks on peripheral edge devices (such as IoT sensors or mobile processors), the physical memory footprint of the serialized model is as critical as its classification accuracy. Extreme reduction of the parameter space is required to fit these models into limited on-chip memory and to reduce energy-consuming off-chip memory accesses [2].

Fig. 1 illustrates the compression trade-off landscape for the ResNet family. The most impressive result of extreme memory reduction is demonstrated by the aggressive hybrid pipeline, Method 09. The sequential application of Two-Stage Distillation, structural pruning, and static INT8 quantization reduces the physical memory footprint from 84.2 MB (baseline teacher, Method 00) to an unprecedented 7.7 MB. This achieves a massive compression ratio of 10.9x. Despite this drastic reduction in capacity, the preserved accuracy stands at 0.789, strictly outperforming all other standalone compact model configurations represented on the chart.

As observed in **Fig. 2**, the massive VGG architecture exhibits an even more pronounced response to hybrid compression. Method 09 successfully reduces the VGG memory footprint from 513.7 MB (baseline teacher) to a mere 74.4 MB. This represents a compression ratio of 6.9x, successfully eliminating the vast majority of redundant parameters stored in the dense fully connected layers.

Throughout this aggressive reduction, the model maintains a highly competitive accuracy of 0.635. It is crucial to note that attempts to achieve similar compression ratios without the proposed Two-Stage Distillation mechanism – such as Method 04, which utilizes standard distillation alongside quantization – yield significantly worse results, dropping accuracy to 0.623. This further validates that advanced distillation is mandatory to preserve the feature space during aggressive 8-bit quantization [6].

Pareto-Efficiency and Cross-Platform Hardware Analysis. The hardware architecture of the target computing device strictly dictates the efficacy of any given compression methodology. A pipeline that performs exceptionally well in a cloud environment may fail entirely on an edge device. To objectively evaluate these trade-offs, Pareto efficiency frontiers were constructed. A compression configuration is considered Pareto-optimal if no other tested method can provide a higher inference speed (Frames Per Second, FPS) without a corresponding degradation in classification accuracy.

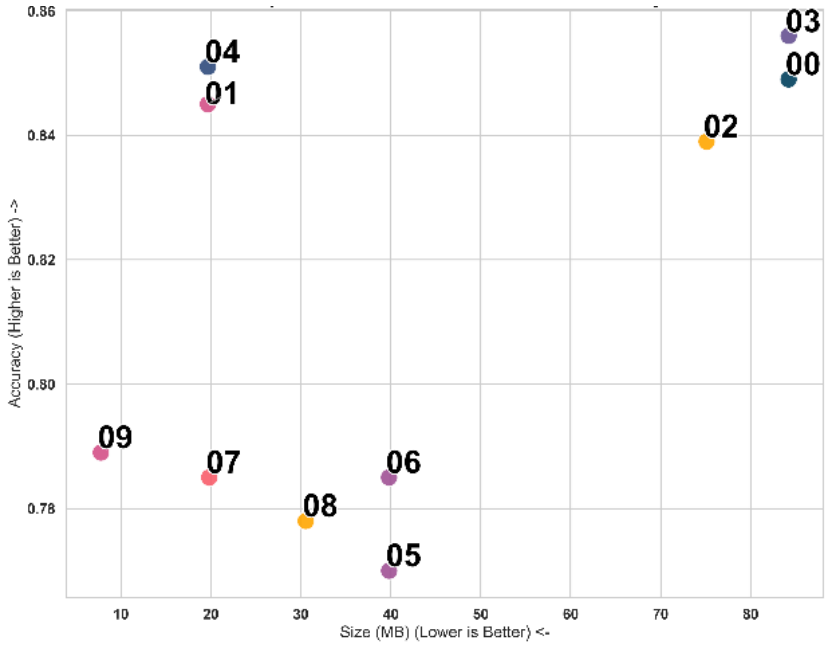


Fig. 1. Trade-off between model size and classification accuracy for the ResNet architecture

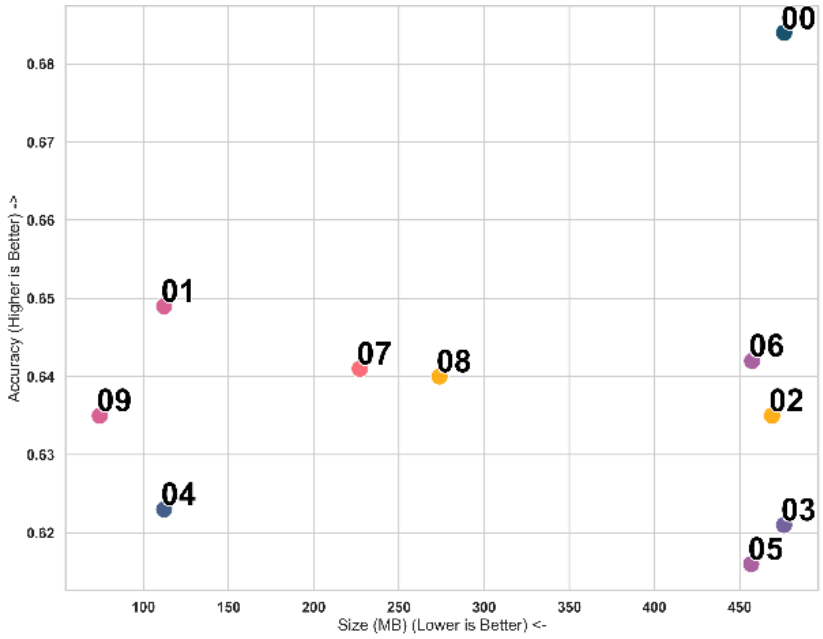


Fig. 2. Trade-off between model size and classification accuracy for the VGG architecture

Fig. 3 visualizes the performance of the VGG architectural family on a Central Processing Unit (CPU), simulating a resource-constrained edge computing environment. The massive, linear structure of VGG traditionally suffers from severe latency on CPUs. The Pareto frontier clearly indicates that Method 09 – the aggressive hybrid pipeline combining Two-Stage Distillation, structural pruning, and static INT8 quantization – is the absolute optimal solution. By translating the 32-bit floating-point weights into 8-bit integers,

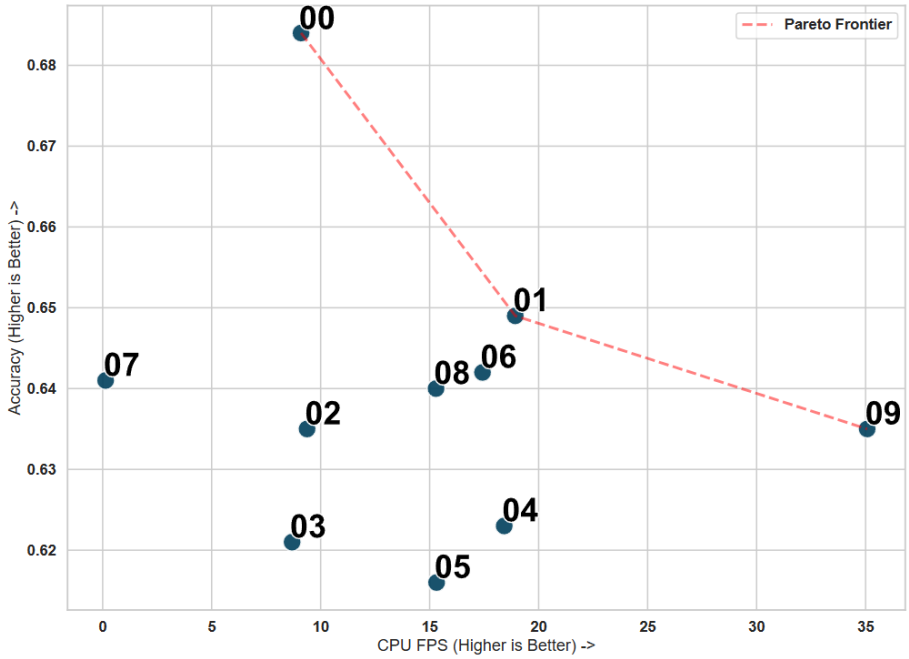


Fig. 3. Pareto efficiency frontier of compression methods for VGG on a CPU platform

the CPU can maximally utilize Advanced Vector Extensions (SIMD instructions) to process multiple operations simultaneously [12].

Consequently, the inference speed increases dramatically from a baseline of 9.09 FPS (Method 00, Teacher) to 35.07 FPS, representing a 3.85x acceleration while maintaining a competitive accuracy of 0.635. Furthermore, the graph demonstrates that isolated static quantization (Method 01) yields only 18.9 FPS, proving that integer arithmetic alone is insufficient for heavy architectures without the topological reduction provided by structural pruning.

The dynamics shift radically when deploying the VGG network on a platform equipped with Compute Unified Device Architecture (CUDA) and specialized Tensor Cores, as illustrated in **Fig. 4**. Notably, all pipelines utilizing INT8 quantization (Methods 01, 04, and 09) are entirely absent from the Pareto frontier. In hardware-accelerated environments, these integer-based methods demonstrate a critical performance collapse, recording 0.0 effective FPS. It is important to emphasize that this observation does not reflect a hardware limitation of modern GPUs but rather the characteristics of the native software environment used in this study. All experiments were conducted exclusively within the standard PyTorch framework without employing specialized inference deployment or optimization frameworks. Under these conditions, PyTorch does not provide fully optimized native execution kernels for direct INT8 inference on GPUs. Within the native PyTorch execution pipeline used in this study, software dequantization and associated memory bandwidth overheads prevent efficient utilization of dedicated GPU integer execution units, resulting in severe performance degradation [16].

Instead, Method 07 dictates the Pareto front. Achieving efficient INT8 inference on modern GPUs typically requires deploying the model through specialized optimization frameworks, such as NVIDIA TensorRT, AMD MIGraphX, or Intel OpenVINO, rather than relying solely on the native PyTorch execution environment used in this work. By converting the distilled student model into a half-precision floating-point format (FP16), the network perfectly aligns with the GPU's native Tensor Core capabilities. This allows the VGG network to be processed at an exceptional 601.78 FPS, outperforming the original teacher

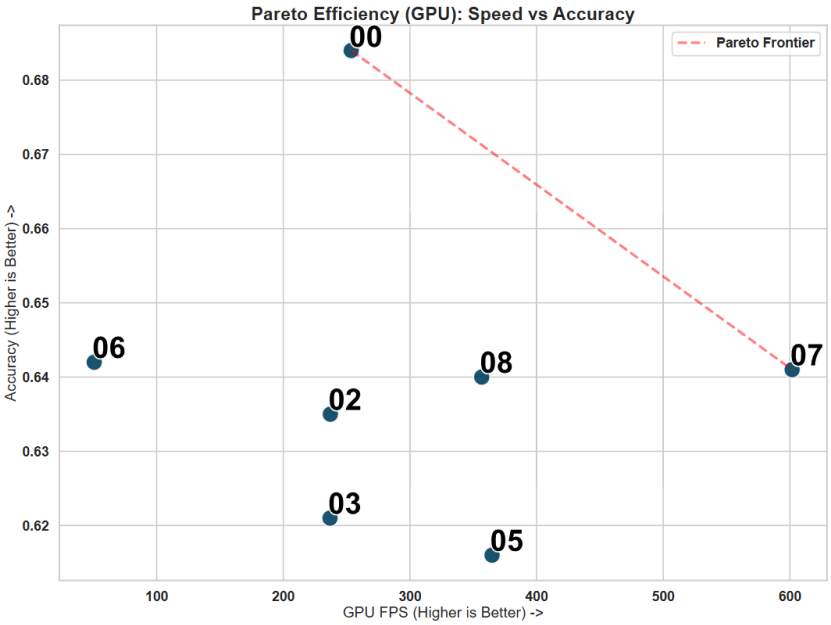


Fig. 4. Pareto efficiency frontier of compression methods for VGG on a GPU platform

(253.6 FPS) by 2.37 times, with minimal accuracy fluctuation (0.641). **Fig. 5** examines the CPU performance of the ResNet family. Inherently more efficient than VGG due to residual connections and the absence of massive fully connected layers, the baseline ResNet50 (Method 00) processes at 23.6 FPS.

However, the Pareto frontier reveals that the Two-Stage Distillation algorithm combined with hybrid compression (Method 09) maximizes hardware efficiency far beyond the baseline. Just as observed with the VGG network, Method 09 forms the optimal peak

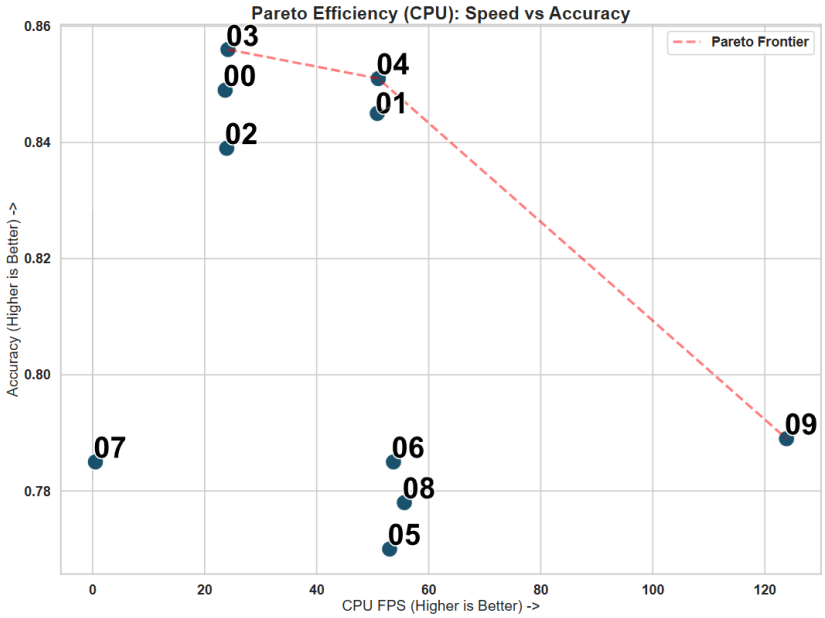


Fig. 5. Pareto efficiency frontier of compression methods for ResNet on a CPU platform

on the CPU. The inference speed leaps from 23.6 FPS to a phenomenal 123.8 FPS, achieving a 5.24x acceleration. Because the accuracy is preserved at 0.789, this specific pipeline proves to be the ideal candidate for deploying real-time, high-accuracy computer vision systems on embedded IoT processors.

Finally, **Fig. 6** demonstrates the performance of the compressed ResNet models on a high-performance GPU. Mirroring the VGG GPU analysis, the application of integer quantization (INT8) leads to zero effective throughput within the native PyTorch environment, completely excluding Methods 01, 04, and 09 from the Pareto frontier. As discussed above, this behavior reflects the limitations of the native software execution environment rather than the hardware capabilities of modern GPUs.

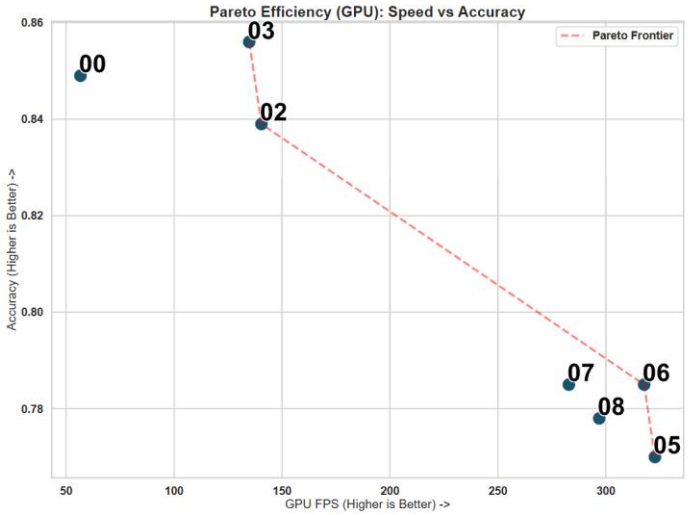


Fig. 6. Pareto efficiency frontier of compression methods for ResNet on a GPU platform

However, the dynamic for ResNet diverges significantly from the VGG family regarding half-precision formatting. On this CUDA-supported platform, Method 07 (FP16) does *not* lie on the Pareto efficiency frontier. Instead, the optimal frontier is formed by pure distillation-based models operating in standard precision: Method 03, Method 02, Method 06, and Method 05. The Two-Stage Distillation applied to the student model in standard FP32 precision (Method 06) achieves a superior balance, reaching an impressive inference speed of approximately 315 FPS with an accuracy of 0.785. In contrast, Method 07 achieves the exact same accuracy but lags in speed (processing at approximately 280 FPS).

This indicates that for inherently optimized, compact architectures like ResNet18, standard 32-bit execution (Method 06) outperforms FP16 conversion, avoiding the hardware overhead associated with precision casting [13]. Method 06 emerges as the definitive Pareto-efficient choice for achieving extremely high throughput on GPUs without compromising the integrity of the distilled feature space.

CONCLUSION

In this research, we have successfully addressed the scientific and practical challenge of enhancing the efficiency of knowledge transfer between convolutional neural networks for deployment on resource-constrained edge devices. Through comprehensive theoretical analysis and extensive experimental modeling across multiple hardware backends, several critical conclusions have been established.

First, the study confirms that standard knowledge distillation protocols are highly vulnerable to the phenomenon of "gradient shock" during the initialization phase,

particularly when applied to linear topologies. For the VGG family, the absence of a stabilization phase resulted in a 6.3% drop in classification accuracy. The implemented Two-Stage Distillation algorithm effectively mitigates this issue by introducing a preliminary stabilization phase. By temporarily freezing the deep convolutional layers and training only the newly initialized classifier, this "warmup" safely isolates and absorbs destructive gradients. This stabilization process consistently improved the accuracy of compact student models, yielding a gain of +2.6% for VGG11 and +1.5% for ResNet18 compared to baseline training from scratch.

Second, the experimental results provide robust evidence for the existence of the "distillation healing" mechanism. We demonstrated that utilizing temperature-smoothed soft targets from a teacher model acts as a powerful spatial navigator for the student's optimizer within a narrowed parameter space. This recovery mechanism allowed structurally pruned architectures to not only mitigate the damage from the removal of nearly 30% of their weights but also to outperform their full-sized baseline student counterparts in accuracy.

Third, the cross-platform analysis established that the optimal compression strategy is strictly hardware-dependent. For edge computing systems (CPU), a hybrid pipeline combining Two-Stage Distillation, structural pruning, and static INT8 quantization emerged as the only viable solution for extreme efficiency. This configuration achieved a 10.9× compression of the physical memory footprint and a 5.24× acceleration of inference throughput while maintaining high generalization capacity.

Finally, for high-performance cloud environments (GPU), the experiments demonstrated that native INT8 inference within the PyTorch framework remained inefficient without specialized inference optimization frameworks. While FP16 formatting remains the Pareto-optimal choice for massive linear architectures like VGG (achieving over 600 FPS), for inherently optimized residual networks like ResNet, standard 32-bit floating-point execution guided by Two-Stage Distillation (Method 06) provides the most efficient balance of speed and accuracy. These findings offer a clear roadmap for the deployment of specialized, hardware-aware AI solutions in real-time computer vision systems.

Despite the obtained results, future research will extend the proposed approach to modern lightweight architectures specifically designed for edge-oriented deployment scenarios. Furthermore, future studies will investigate fully optimized inference pipelines based on specialized vendor-specific frameworks and graph compilers, such as NVIDIA TensorRT, AMD MIGraphX, and Intel OpenVINO, to provide a more comprehensive assessment of INT8 quantization efficiency on dedicated hardware accelerators. Such extensions will enable a broader evaluation of hardware-aware compression strategies beyond the native framework execution environment considered in this study.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author(s) received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no conflicts of interest relevant to this study.

AUTHOR CONTRIBUTIONS

Conceptualization, [A.K., L.D.]; methodology, [A.K.]; validation, [A.K., L.D.]; formal analysis, [A.K.]; investigation, [A.K.]; resources, [A.K.]; data curation, [A.K.]; writing – original draft preparation, [A.K.]; writing – review and editing, [A.K., L.D.]; visualization, [A.K.]; supervision, [A.K., L.D.]; project administration, [A.K., L.D].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11976-11986). https://doi.org/10.1109/CVPR52688.2022.01167

[2] Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020). Model compression and hardware acceleration for neural networks: A comprehensive survey. In *Proceedings of the IEEE*, 108(4), 485-532. https://doi.org/10.1109/JPROC.2020.2976475

[3] Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6), 1789-1819. https://doi.org/10.1007/s11263-021-01453-z

[4] Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A. A., & Wilson, A. G. (2021). Does knowledge distillation really work? In *Advances in Neural Information Processing Systems (NeurIPS)* (Vol. 34, pp. 6906-6919). https://arxiv.org/abs/2106.05945

[5] Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., & Guttag, J. (2020). What is the state of neural network pruning?. *Proceedings of Machine Learning and Systems (MLSys)*, 2, 129-146. https://arxiv.org/abs/2003.03033

[6] Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., Van Baalen, M., & Blankevoort, T. (2021). A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*. https://arxiv.org/abs/2106.08295

[7] Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2022). A survey of quantization methods for efficient neural network inference. In G. K. Thiruvathukal, Y.-H. Lu, J. Kim, Y. Chen, & B. Chen (Eds.), *Low-power computer vision: Improve the efficiency of artificial intelligence* (pp. 291–326). Chapman and Hall/CRC. https://doi.org/10.1201/9781003162810-13

[8] Ngo, T. T., Nguyen, B., Pham, T. B., Truong, S. N., Vo, H. M., Minh, K.-S., & Pham, K. V. (2026). Progressive compression of ResNet: An 8.33× reduction with 90.5% accuracy retention for edge deployment. *IEEE Access,* 14, 56900–56918. https://doi.org/10.1109/ACCESS.2026.3682854

[9] Lai, N., Dewi, D. A., Maidin, S. S., Xiao, W., Zhao, S., & Hu, Q. (2026). A comprehensive review of lightweight deep learning models for edge computing with future directions. *Discover Computing,* 29, 110. https://doi.org/10.1007/s10791-026-10021-3

[10] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 1-74. https://doi.org/10.1186/s40537-021-00444-8

[11] Liebenwein, L., Baykal, C., Lang, H., Feldman, D., & Rus, D. (2020). Provable filter pruning for efficient neural networks. In *International Conference on Learning Representations.* https://arxiv.org/abs/1911.07412

[12] Wu, H., Judd, P., Zhang, X., Isaev, M., & Micikevicius, P. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*. https://arxiv.org/abs/2004.09602

[13] Yun, J., Kang, B & Fu, Z. (2023). The hidden power of pure 16-bit floating-point neural networks. *arXiv preprint arXiv:2301.12809*. https://arxiv.org/abs/2301.12809

[14] Gyawali, D. (2023). Comparative analysis of CPU and GPU profiling for deep learning models. *arXiv preprint arXiv:2309.02521*. https://arxiv.org/abs/2309.02521

[15] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style ConvNets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13733-13742) https://doi.org/10.1109/CVPR46437.2021.01352

[16] Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., ... & Keutzer, K. (2021). HAWQV3: Dyadic neural network quantization. In *International Conference on Machine Learning* (pp. 11875-11886). https://arxiv.org/abs/2011.10680

АПАРАТНО-ОРІЄНТОВАНА ГІБРИДНА КОМПРЕСІЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ СИСТЕМ З ОБМЕЖЕНИМИ РЕСУРСАМИ

Андріан Козинець, Лідія Демків
Кафедра системного проектування,
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, м. Львів, 79005, УКРАЇНА

АНОТАЦІЯ

Вступ. Глибокі нейронні мережі стикаються з критичними обчислювальними бар'єрами для розгортання на периферійних пристроях з обмеженими ресурсами. Хоча існують окремі методи компресії, їх ізольоване застосування спричиняє значне зниження точності. Синергетичне поєднання обрізання (прунінгу), квантування та дистиляції знань пропонує рішення, проте їхня взаємодія та апаратно-залежна ефективність залишаються недостатньо дослідженими.

Матеріали та методи. Це дослідження розглядає гібридні конвеєри компресії, застосовані до архітектур згорткових нейронних мереж із послідовним з'єднанням шарів та архітектур із залишковими зв'язками. Оцінено алгоритм двоетапної дистиляції з додатковою попередньою фазою стабілізації класифікатора для усунення градієнтного шоку в поєднанні зі структурним видаленням фільтрів та статичним цілочисельним квантуванням. Кросплатформний аналіз проводився на центральних і графічних процесорах із побудовою меж ефективності Парето для визначення оптимальних компромісів.

Результати. Фаза попередньої стабілізації успішно усунула градієнтний шок, підвищивши точність моделей-учнів на величину до 2.6%. Ізольоване структурне обрізання значно погіршило точність, проте «лікування дистиляцією» за допомогою м'яких цілей учительської моделі дозволило обрізаній мережі (із видаленням майже 30% ваг) перевершити свою повнорозмірну базову лінію. На центральних процесорах (ЦП) гібридний конвеєр зменшив обсяг пам'яті в 11 разів і прискорив виведення у 5 разів. Натомість цілочисельне квантування виявилося неефективним на графічних процесорах (ГП) через затримки доступу до пам'яті. Для таких прискорювачів формат чисел із плаваючою комою половинної точності був Парето-оптимальним для масивних архітектур із послідовним з'єднанням шарів, тоді як формат чисел із плаваючою комою одинарної точності в поєднанні з двоетапною дистиляцією забезпечив найкращу ефективність для архітектур із залишковими зв'язками.

Висновки. Інтеграція двоетапної дистиляції знань зі структурним обрізанням та квантуванням принципово перевершує ізольовані методи компресії. Оптимальна стратегія суворо залежить від апаратного забезпечення та топології: цілочисельні гібридні конвеєри є необхідними для енергоефективних периферійних ЦП, тоді як на ГП вибір між половинною та стандартною точністю визначається структурними характеристиками конкретної мережі.

Ключові слова: Штучний інтелект, Глибоке навчання, Оптимізація нейронних мереж, Системи з обмеженими ресурсами, Апаратне прискорення.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
10 June, 2026	08 July, 2026	10 July, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 31–44
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.6, 004.89, 004.932, 81`32

COMPRESSION COEFFICIENT IN THE ANALYSIS OF NOISY TEXTS AND IMAGES

Oleksii Kravchuk* & ***Oleh Kushnir***
Department of Optoelectronics and Information Technologies,
Ivan Franko National University of Lviv,
107 Gen. Tarnavskoho Street, Lviv, 79017, UKRAINE

Kravchuk, O. M., & Kushnir, O. S. (2026). Compression Coefficient in the Analysis of Noisy Texts and Images. *Electronics and Information Technologies*, 35, 31–44. https://doi.org/10.30970/eli.35.3

ABSTRACT

Background. We analyze the compression coefficient S associated with the two versions of a Lempel–Ziv algorithm (LZA) as a quantitative characteristic of structural order in noisy or corrupted data, using a natural-language text and a digital image. To model a stochastically perturbed text or image, we randomize the linguistic elements of the former and add uncorrelated Gaussian noise to the latter or randomize its pixels.

Materials and Methods. A fiction text "Moby Dick" by H. Melville was randomized at the linguistic levels of sentences, words, and characters. The number of randomization cycles N varied in the range from 0 to 10^9. We also analyzed the red channel of an 8-bit image "Peppers" with the size $M = 512\times512$. The Gaussian noise with the standard deviations σ changing in the range from 0 to 400 was added to the image. On the whole, 50 randomized and 50 noisy image versions were generated. To apply the LZA compression to the image, we unfolded it into a one-dimensional sequence of pixel intensities and processed the appropriate textual representation of the image as usual text.

Results and Discussion. The sentence-level randomization almost does not change the S value, because it remains intact the internal structures of words and sentences, while the LZA has a local character. The word-level randomization decreases S, whereas the character-level randomization leads to the largest S decrease and produces almost incompressible text files at the largest N. The results obtained with the two LZA-type versions differ slightly in the effect of sentence randomization on the compressibility. For the randomized images, the S value sharply decreases when N approaches the total pixel number M, and then reaches saturation. The strongest changes in S for the noisy images occur in the range of $\sigma \in [10, 100]$.

Conclusion. Our results demonstrate that the LZA compression coefficient is sensitive to the loss of local order in the noisy data of different types, either randomized texts or randomized and noisy images. Since the LZA is based on repetitions and local regularities in a symbolic sequence, a decrease in S indicates partial destruction of order caused by randomization or noise. Hence, the compression coefficient can, in principle, serve as a measure of disorder in both symbolic sequences and pixel grids and so can help in detecting noisy data.

Keywords: data compression, noisy data analysis, noise detection, randomization, natural language processing, image processing.

1. INTRODUCTION

There are many practical situations when the data of interested for information retrieval is noisy or partly corrupted. Among diverse examples, one can recall, in this

© 2026 Oleksii Kravchuk & Oleh Kushnir. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

respect, mangled or damaged textual documents, visual or compression artifacts of digital images, grainy images, corrupted footage, or damaged frames. As a limiting case, the data can become so corrupted that its information content cannot be retrieved against the background of dominating noise. Then the problem is to recognize this noisy data from the neat information-carrying one, using no human intervention or visual inspection.

The main point of interest in the present work is the possibilities for detection of 'noise' in textual documents. This can be done, e.g., using different complexity measures and some other approaches [1–6]. Technically, a text represents a sequence – or time series – of linguistic symbols. Our auxiliary point is therefore to find such statistical properties or characteristics that can help in differentiating between the informative and (partly) noisy symbolic time series. One of the possible characteristics is the compressibility of a text file [7–11]. Here the motivation is simple: ordered and information-bearing sequences are usually more compressible than noisy or randomized ones [1, 3, 12–14]. Indeed, any lossless compression algorithm finds repeated fragments and regularities in the input data. If the noise destroys these regularities, the size of the compressed file increases and the compression coefficient decreases.

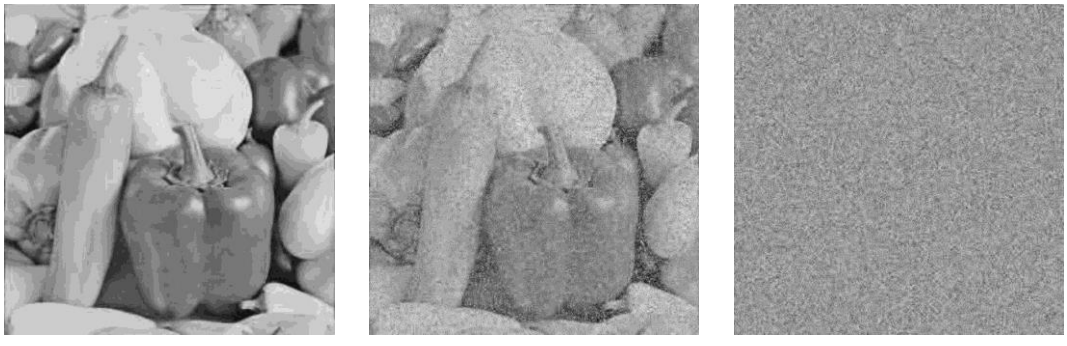
In other words, it is important that the compression coefficient reacts not only to the frequencies of linguistic elements given by Zipf's or Heaps' laws (see Ref. [15]), but also to their order. The other point of our interest is automatic detection of noisy images [16–20]. It is important that the compression-based approach can, in principle, be applied to images, since the latter can be viewed as grids of pixel values [20, 21].

The aim of this study is to analyze the compression coefficient for disordered or corrupted natural-language texts and noisy images. As a model of stochastically perturbed text, we choose the simplest mechanism of partial randomization (or shuffling) of its linguistic elements such as sentences, words, and characters. To model a partially corrupted image, we add uncorrelated Gaussian noise to it and, moreover, randomize its pixels. To analyze the changes occurring with a text or image under noising, we examine a gradual evolution of its compression coefficient during the noising process.

2. MATERIALS AND METHODS

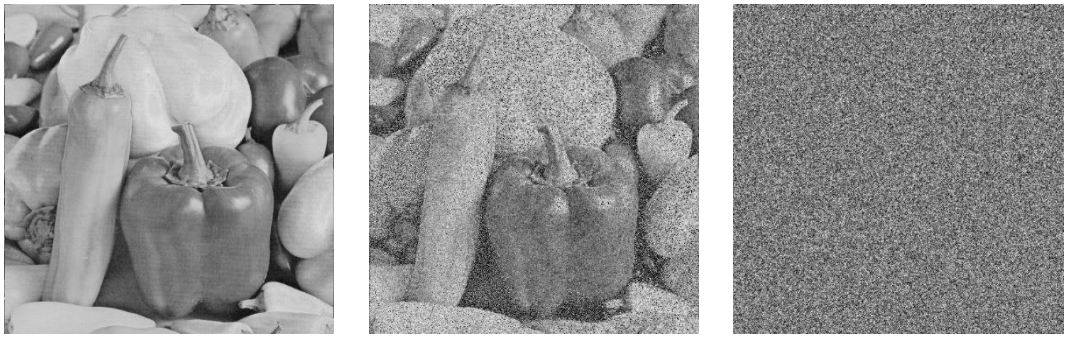
2.1. Data under study and noising procedures

The objects of our study are a natural-language text "Moby Dick" by H. Melville and a digital image "Peppers" which is often analyzed in the field of image processing (see **Fig. 1** and **Fig. 2**). Notice that the text under test includes approximately 1.24×10^6 characters, 2.12×10^5 words and 1.01×10^4 sentences. The 8-bit color image «Peppers» stored in the JPEG format has the size $M = 512\times512$ pixels.



(a) (b) (c)

Fig. 1. Examples of randomization results obtained for the red channel of the image "Peppers" at the randomization cycle numbers $N = 0$ (original image) (a), $N = 73114$ (b), and $N = 10^8$ (c)



(a) (b) (c)

Fig. 2. Examples of Gaussian noise results obtained for the red channel of the image “Peppers” at the noise parameters $\sigma = 0$ (original image) (a), $\sigma = 35.6$ (b), and $\sigma = 402.4$ (c)

As already mentioned above, we modeled corruption of texts in the simplest manner, as partial randomization of their linguistic elements. The starting point of our experimental procedures was the assumption that the compressibility of a text as a symbolic sequence should reflect the amount of order preserved in it. Randomization was treated as a multiply repeated pairwise exchange of randomly selected linguistic elements of a text. The degree of randomization was measured by the number N of elementary randomization cycles, where one cycle corresponded to a single mutual exchange of two elements. The number of cycles was varied in the range from $N = 0$ (the original text) up to $N = 109$. This was performed in 50 discrete steps. Note that the maximal cycle number $N_{\max}$ was far larger than the total number L of its linguistic elements (either characters or words, or sentences) in our text (see the figures mentioned above).

The text was randomized separately at the linguistic levels of sentences, words, and characters. As a result, three randomized datasets were obtained for the initial text. The type of element being exchanged depended on the selected linguistic level of randomization. During the sentence-, word-, and character-level randomizations, whole sentences, individual words, and separate characters were permuted, respectively.

In the same way, the pixels of our image were randomized such that, during each of the elementary randomization cycles, the pixel values were left unchanged, but the spatial locations of the pixels were swapped. The quantitative parameter governing image corruption was again the number N of randomization cycles, which was changed from 0 to 10^8. Here the relation $N_{\max} >> M$ held true. Besides, Gaussian noise with a zero mean was added to the image. It was dosed by the appropriate standard deviation σ, which was increased exponentially from 0 to 400 in order to cover the regions of both weak and strong image perturbations. Notice that the standard deviation characterizing the pixel-value variation in the unnoised image was equal to $\sigma_0 \approx 66.1$, so that the maximal σ applied was surely much larger than the σ_0 value. As a consequence, we obtained 50 noisy and 50 randomized image versions. Some examples of the randomized and noisy images generated as explained above are shown respectively in **Fig. 1** and **Fig. 2**.

2.2. Lempel–Ziv compressibility

The text file was compressed using a freely available Lempel–Ziv algorithm (LZA). The LZA family comprises lossless compression methods that exploit repetitions in the input data and require no predefined statistical model for the data [7]. Instead of assigning short codes only to frequent individual symbols, these methods identify recurring sequences in the data and replace later occurrences with the references that a decoder can interpret. This principle allows the compressor to adapt to the structure of the processed data and capture regularities at different scales, including recurring symbol

sequences or appropriate bit or byte patterns which correspond to characters, words, or still longer textual elements.

The present work employs two specific realizations of the LZA family. The first uses a dictionary-coding strategy intended primarily for symbolic strings, while the second operates with a byte-oriented format. Although both methods are lossless, they should not be treated as interchangeable implementations of a single algorithm, because they discover and encode redundancy using different internal structures and are designed for different operational purposes.

In the frame of the first compression method, the input data is viewed as a sequence of symbols. During a left-to-right scan, the encoder searches for the longest sequence already represented in its current dictionary. The code of that sequence is emitted, and a longer sequence formed by appending the next input symbol is inserted into the dictionary whenever it has not occurred before. On the contrary, the second method operates on byte sequences. Its smallest input unit is a complete byte rather than an individual bit or a linguistic token. At each position, the encoder searches for previously processed bytes for a sufficiently long sequence matching the bytes beginning at the current position. When a match is found, the repeated bytes are replaced by a back-reference that consists of an offset and a match length. The offset specifies how many bytes backward the previous occurrence begins, while the length specifies how many bytes the decoder must copy. Notice that still different LZA-type compression techniques are known from the literature, e.g. that operating on the bit level of symbolic sequences (see, e.g., Ref. [14]).

For every original and randomized text version, the compression coefficient S was calculated as the relative decrease in the file size occurring after compression:

$$S = 1 - s_{out}/s_{in},$$

where s_{in} and s_{out} are the sizes of the original and compressed text files, respectively. Notice that the results obtained with both algorithms depend on the repeatability of text fragments and local regularities in the input symbolic sequence. Then large S values correspond to well-compressible ordered sequences, whereas the S values close to zero imply that the compressed file is almost as large as the input one.

Notice that the LZA is directly fit for the one-dimensional symbolic sequences. Of course, there is a great number of compression techniques aimed at two-dimensional images [16]. However, in order to adhere to the same experimental compressibility measure in the case of two-dimensional digital images and compare the appropriate results with those obtained for the texts, one can conventionally reduce the images to linear chains of pixel values. Simple methods for this aim have been suggested in the earlier works [20, 21]. The main idea of those works was to transform a two-dimensional image into a one-dimensional sequence of pixel values, using a horizontal or vertical scan, and then analyze this sequence by the methods of computational linguistics applied to texts.

In practice, an image can be converted to a linear chain of pixel values by a row-by-row or column-by-column scanning. In this representation, the figures corresponding to pixel intensities play the role of ‘words’ in a text. Although failing to reflect all the two-dimensional neighborhoods in an input image correctly, the image-to-text transformation makes it possible to consider the image properties within the same framework based upon the analysis of one-dimensional symbolic sequences. To be specific, our image was unfolded into a text along the horizontal direction. For this purpose, we used the data of the red channel of the color image.

Finally, an additional check-up compression-related analysis was performed immediately with the JPEG files of the input image and its noised and randomized versions. Here we directly compared the sizes of the files, with no intermediate conversion of image files into text files and with no LZA analysis. The compression parameter K was defined as

$$K = s_0/s,$$

where s_0 and s denote the JPEG-file sizes of respectively the input and noised (perturbed) images. The K coefficient was calculated for both noisy and randomized image datasets. This analysis is not identical to the calculation of S, because a lossy JPEG format includes its own compression procedures. Nevertheless, it provides an additional check of how noising and randomization affect the practical compressibility of the image file.

3. RESULTS AND DISCUSSION

3.1. Randomized texts

Fig. 3 shows the dependence of the compression coefficient S on the randomization parameter N for the three linguistic levels of randomization. The panels (a) and (b) correspond respectively to the first and second LZA-type compression methods (see Subsection 2.1). Here the randomization versions that correspond to the sentence, word, and character linguistic levels are denoted by the curves of different colors (see the legend in **Fig. 3**).

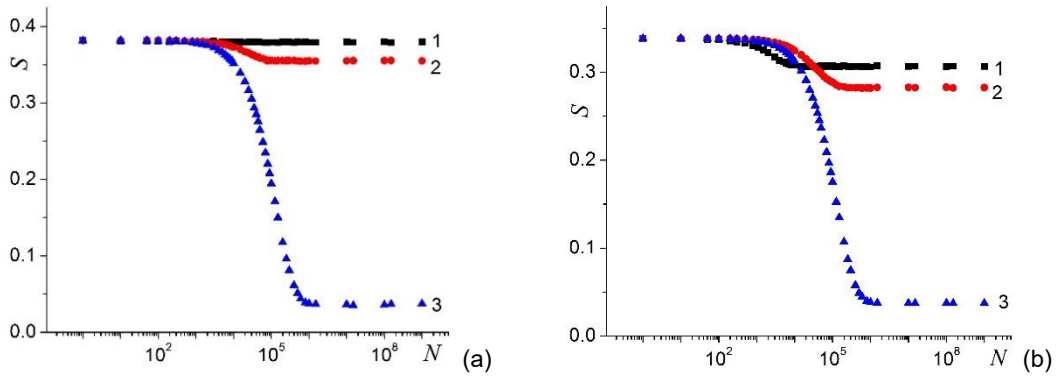


Fig. 3. Dependences of compression coefficient S on the randomization parameter N obtained for the text "Moby Dick". Panels (a) and (b) correspond to different versions of LZA-type libraries (respectively, character-level and byte-level compressions – see the text). Data points denoted by black squares (curves 1), red circles (curves 2), and blue triangles correspond, respectively, to randomization performed at the levels of sentences, words, and characters. For visualization on the logarithmic scale, the initial data points corresponding to $N = 0$ are conventionally represented by $N = 1$

The first important result is the almost complete absence of changes in the compressibility observed under the sentence-level randomization (see curve 1 in **Fig. 3** and data in **Table 1**). Note that the notation L in **Table 1** implies the text length, S_{in} is the compressibility S measured at $N = 0$, S_{avg} is the S value averaged over the last 10 data points for strongly randomized texts, and ΔS is defined as $\Delta S = S_{in} - S_{avg}$. To be exact, the changes mentioned above are negligible in the case of character-level compression and amount to only 10% of the changes observed under character-level randomization in the case of byte-level compression. This means that changing the order of sentences does not have a significant effect on the file compressibility. The reason is that the LZA algorithm applied by us has an essentially short-range character: its result depends on the order of linguistic elements (or their binary representations) located close to each other. In other words, the sentence-level permutations change the global structure of the text radically, but most of the local structures inside separate sentences remain invariable. Notice that the reasons for different results of the two alternative LZA realizations associated with the

Table 1. **Some data related to the compression coefficient obtained for the original and randomized texts, using the character-level LZA compression**

Quantitative characteristic	Sentence-level randomization	Word-level randomization	Character-level randomization
Text length L	10,076	212,288	1,239,464
Compressibility values S_{in} and S_{avg} before and after randomization	0.381 → 0.379	0.381 → 0.355	0.381 → 0.039
Compressibility change ΔS under randomization	0.002	0.026	0.342

Note: The data in the third row of the table imply $S_{in} \to S_{avg}$.

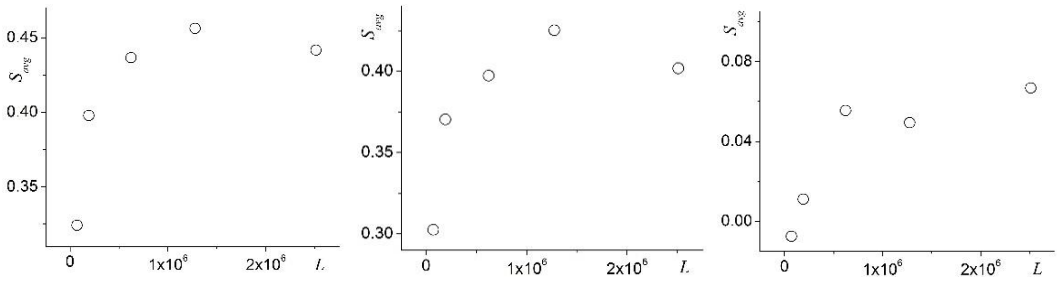
character and byte levels deserve further study. Since the second LZA realization manifested notably higher data point scattering in case of image-mapping compression, we will consider only the first LZA realization for images.

A notably different behavior is observed for the word-level and, especially, character-level randomization (see curves 2 and 3 in **Fig. 3a** and **Fig. 3b**). In both cases, the compression coefficient decreases notably with increasing N. This decrease reflects the gradual destruction of local regularities in the symbolic sequences. In particular, the word-level randomization destroys the order of words, though the internal structure of each word remains intact. Therefore, the limiting S value at the largest N remains relatively high. Finally, the character-level randomization produces the strongest effect because it destroys not only the word order but also the internal structure of words.

The other important fact is that an expressed drop in the S parameter is observed whenever the cycle number N becomes $N \approx L$ or somewhat larger, with L denoting the total number of sentences, words, or characters in the text. This can be seen both in **Fig. 3a** and, especially, in **Fig. 3b** where the effect of the sentence-level randomization is not negligible. Indeed, the S values referred to the sentence-, word-, and character-level randomization are stabilized after dropping at approximately $N \approx 10^4$, $2{\times}10^5$, and $1.2{\times}10^6$, respectively. A comparison with the L data indicated in Table 1 confirms that $N \approx L$ really represents the condition for stabilization of the S parameter. At $N \gg L$, the symbolic sequence is practically fully randomized, and any further increase in N produces no noticeable changes. In this limiting case, the S parameter approaches zero in the most radical regime of character-level randomization, so that we have $S_{in} \approx S_{avg}$ (see the S_{avg} value in the last column of **Table 1**), which means that the text file becomes almost incompressible.

Thus, the total difference ΔS between the S values found for the original text and its strongly randomized versions depends strongly on the randomization level. It is maximal for the character-level randomization, smaller for the word-level randomization, and almost absent for the sentence-level randomization. This confirms that the compression coefficient is much more sensitive to the local order than to the global one.

It would be convenient if the difference $\Delta S = S_{in} - S_{avg}$ between the compression coefficients of unnoised and strongly noised texts were a function of only randomization type. Then a large ΔS parameter would have served as a criterion of strong text-data corruption. However, this hypothesis needs examining if ΔS does not depend on the text size. To clarify whether the limiting compressibility S_{avg} of strongly randomized texts depends on the text size L, we have measured the $S(N)$ dependences for the four additional natural texts of English-language fiction with essentially different lengths L and then calculated the averaged parameters S_{avg}. The results are shown in **Fig. 4** for the three



(a) (b) (c)

Fig. 4. Dependence of the limiting compression coefficient value S_{avg} of strongly randomized texts, which is averaged over the last 10 points of the appropriate $S(N)$ curves, on the text size L expressed in terms of character number. Panels (a), (b) and (c) correspond respectively to the S data obtained at the sentence, word and character randomization levels. The data refers to the character-level version of LZA

linguistic levels of randomization. Note that the first data point $S_{avg} < 0$ in **Fig. 4c** obtained for the shortest text is merely an experimental artifact that reflects nothing but extremely small S values typical for the short random symbol sequences.

For all the linguistic levels of randomization, we observe an almost monotonous, increasing dependence $S_{avg}(L)$ that tends to become saturated at the largest L. In other words, the noised texts of larger sizes usually have larger compression coefficients. This indicates that the limiting compressibility of strongly randomized texts is determined, first of all, by the number of their linguistic elements. The appropriate reason is natural: the gain in the file size due to compression becomes larger for larger files. Besides, the data point scattering observed in **Fig. 4** testifies that some other properties of texts also influence the S_{avg} value. Since no internal local order or the corresponding regularities of repetitions are peculiar to the totally randomized texts, the properties mentioned above can be the alphabet and vocabulary sizes of the original text, as well as the frequency distributions of its linguistic elements (characters or words).

It is seen from **Fig. 4** that, unfortunately, the S_{avg} parameter is indeed affected by the text size, so that the absolute ΔS value cannot be directly used to judge if an unknown text is strongly noised. Nevertheless, one witnesses a gradual saturation of the $S_{avg}(L)$ functions with increasing L around some limiting values. The remaining shortcoming of this situation is that the corresponding lengths L are relatively large, thus preventing one from any categorical ΔS-based conclusions about the noising degree for short textual documents.

3.2. Randomized and noisy images

The dependences of the compression coefficient of the image “Peppers” on the randomization and noise parameters are shown in **Fig. 5**. For the randomized images, the compression coefficient sharply decreases when the number of randomization cycles approaches the value $M = 512\times512 \approx 2.6\times10^5$ (see **Fig. 5a**). Further increase in N has almost no influence on S. This behavior is analogous to the case of texts: the characteristic number of random permutations needed to destroy the order in the symbolic sequence is of the same order of magnitude as the number of elements in this sequence.

On the other hand, the dependence for the noisy images (see **Fig. 5b**) is controlled by the standard deviation σ of the Gaussian noise. The strongest S change occurs approximately in the interval $\sigma \in [10; 100]$. This interval is close to the standard deviation σ_0 of the pixel values in the original image ($\sigma_0 \approx 66$ – see Subsection 2.1). Therefore, the main loss of compressibility occurs whenever the external noise becomes comparable with the ‘internal’ pixel-level variations in the original image.

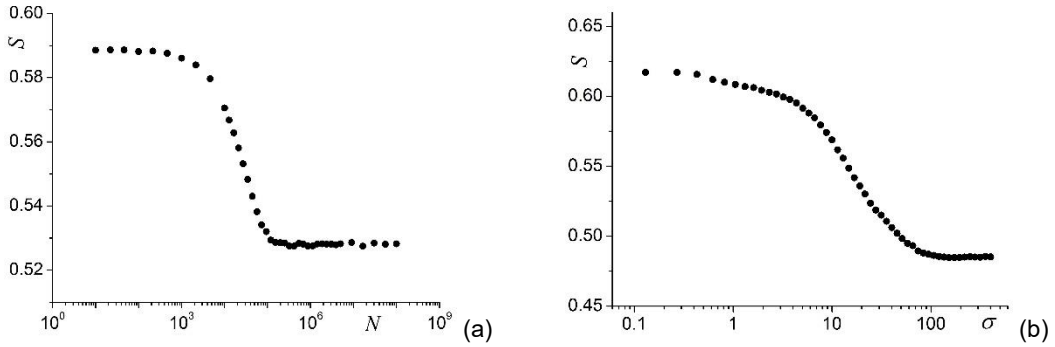


Fig. 5. Dependence of compression coefficient S on the randomization parameter N (a) and the Gaussian noise parameter σ (b), as obtained for the image "Peppers". The data refers to the character-level version of LZA. For visualization on the logarithmic scale, the initial data point in panel (a) corresponding to $N = 0$ is conventionally represented by $N = 1$

The minimal compression coefficients for our image are far from zero and notably higher than the S_{avg} values obtained for the word-randomized, not to say character-randomized texts. This is seen from **Table 2**. Here S_{in} denotes the compressibility S measured at $N = 0$ or $\sigma = 0$, S_{avg} the S value averaged over the last 10 data points, and ΔS is defined as $\Delta S = S_{in} - S_{avg}$. This is explained by a specific structure of textual representations of images. Even a strongly randomized or noised image preserves significant local regularities associated with the 'internal structure' of the numbers corresponding to pixel levels, the limited 'vocabulary' of 256 possible pixel values (0 to 255), frequent random repetitions of these values, and additional restrictions imposed by the intensity histograms.

Table 2. **Details of the compression coefficient obtained for the original, randomized, and noised images, using the character-level LZA compression**

Quantitative characteristic	Pixel randomization	Gaussian noising
Compressibility values S_{in} and S_{avg} before and after randomization or noising	0.589 → 0.528	0.589 → 0.485
Compressibility change ΔS under randomization or noising	0.061	0.104

Note: The data in the second row of the table imply $S_{in} \to S_{avg}$.

The image "Peppers" has been additionally analyzed without intermediate conversion into a one-dimensional text file. Then the characteristic parameter K is calculated simply from the sizes of JPEG files (see Subsection 2.2). The corresponding dependences are shown in **Fig. 6** (see also **Table 3**). Note that the notation K_{in} in **Table 3** refers to $N = 0$ or $\sigma = 0$, K_{avg} is the K value averaged over the last 10 data points, and ΔK is defined as $\Delta K = K_{in} - K_{avg}$. The dependence $K(N)$ in **Fig. 6a** for the randomized images looks as expected: as the randomization degree increases, the spatial structure of the image is destroyed and the JPEG representation becomes less efficient. Consequently, the file sizes of the randomized images increase and the corresponding size ratio K decreases. In general, the characters of the $K(N)$ and $K(\sigma)$ functions are similar: after a transition region, both functions approach the limiting values that correspond to strongly disordered images.

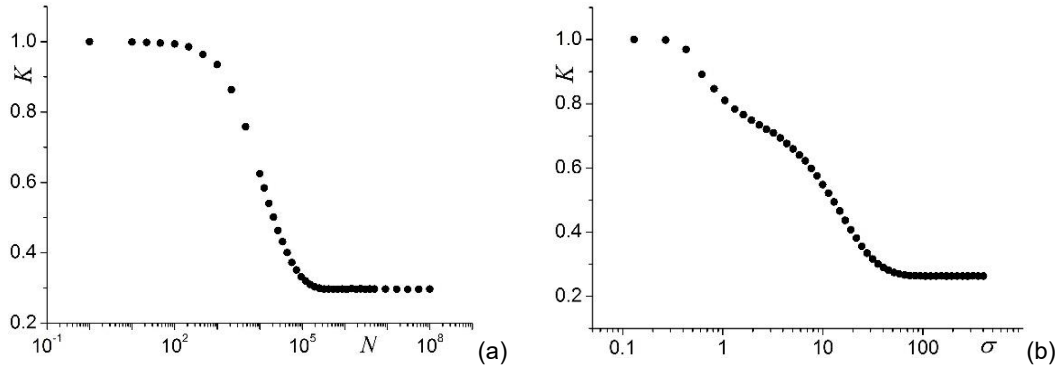


Fig. 6. Dependence of the compression parameter *K* associated with JPEG file sizes of the image "Peppers" on the randomization parameter N (a) and the Gaussian noise parameter σ (b). For visualization on the logarithmic scale, the initial data point in panel (a) corresponding to $N = 0$ is conventionally represented by $N = 1$

Table 3. **Details of the JPEG size-based compression parameter *K* obtained for the randomized and noised images**

Quantitative characteristic	Pixel randomization	Gaussian noising
Compressibility parameters K_{in} and K_{avg} before and after randomization	1.00 → 0.30	1.00 → 0.26
Change ΔK in the compressibility parameter K under randomization or noising	0.70	0.74

Note: The data in the second row of the table imply $K_{in} \rightarrow K_{avg}$.

3.3. General data interpretation

The results obtained by us for both a sample text and a sample image are consistent with each other. In particular, the compression coefficient decreases when the structural order in the symbolic time series is destroyed. The clearest analogy is observed between the word-level text randomization and the pixel-intensity randomization in an image. In both cases, the elementary 'semantics-carrying' elements (respectively words and pixel values) are permuted, and a sharp change appears when the number of permutations becomes comparable with the total number of the elements.

A weak influence of the sentence-level randomization on the text compressibility should not be interpreted as evidencing that the sentence order is unimportant. Rather, this demonstrates that the compression-based method is mainly sensitive to the local regularities. In this respect, our method is close to other short-range characteristics, like the network ones [21], and should be used for quantifying local order but not global semantic coherence.

In order to better understand the behavior of compressibility depending on N and σ, it would be convenient to study some alternative characteristics of the structural disorder in digital images, statistical or other measures of complexity, and compare them with those described above. For this aim, we have calculated the information Shannon entropy *SE* and the peak signal-to-noise ratio *PSNR* for the image "Peppers" (see **Fig. 7**) [16].

As already noticed above, the *SE* parameter $H = -\Sigma_k\, p_k \log_2 p_k$ calculated from the pixel-value probability distribution p_k (i.e., the image histogram) is an invariant of the pixel randomization procedure and reveals no changes (**Fig. 7a**). In other words, the LZA compression for images in this respect is superior if compared to the histogram-based

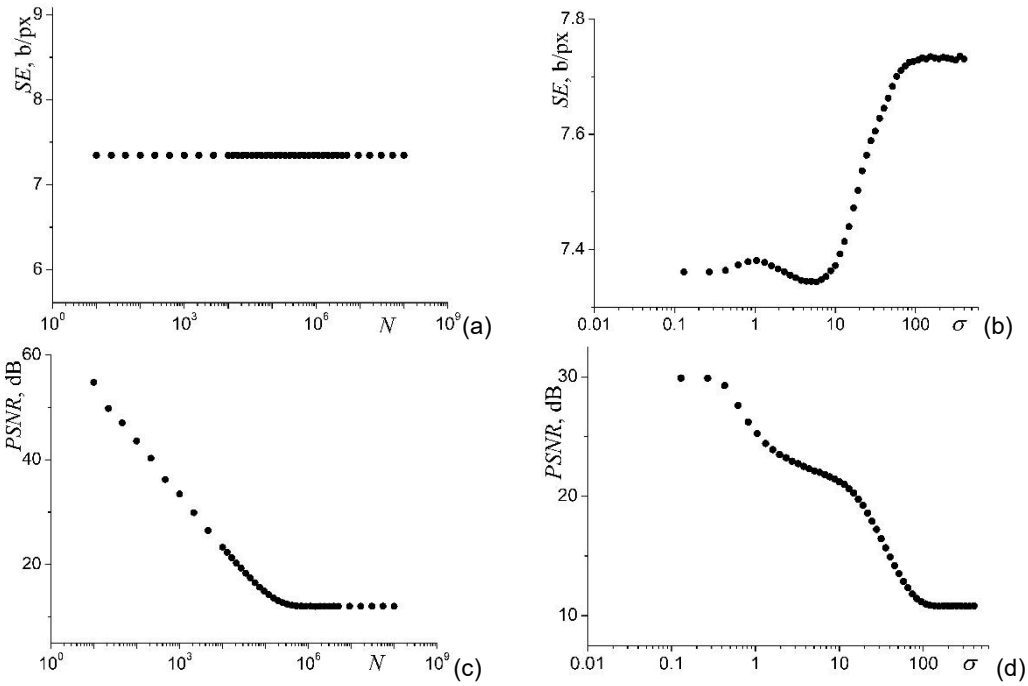


Fig. 7. Dependence of Shannon entropy *SE* (a, b) and *PSNR* parameter (c, d) of the image "Peppers" on the randomization parameter *N* (a, c) and the Gaussian noise parameter *σ* (b, d)

parameters, which are insensitive to the disorder induced by pixel shuffling. The dependence *SE*(σ) shown in **Fig. 7b** can easily be explained: the uncertainty of the p_k distribution increases with increasing noise parameter σ.

Finally, the dependences *PSNR*(N) and *PSNR*(σ) in **Fig. 7c, d** are much smoother in comparison with *SE*(σ). They are qualitatively similar to the dependences $S(\sigma)$ (Fig. 5b) and, especially, $K(\sigma)$ (Fig. 6b). In particular, the *PSNR*(N) function reveals no initial plateau at the smallest N, unlike the dependences $S(N)$ and $K(N)$. The corresponding reasons for so different behaviors need further analysis.

Overall, the main patterns observed by us agree well with the general considerations about the compressibility of nonrandom data [1, 3, 12, 13]. They also indicate a possible practical direction: the compression-based parameters can be used as simple indicators of disorder – due to either randomization or noise – which is present in the symbolic sequences of different natures.

4. CONCLUSIONS

We have modeled stochastically perturbed or corrupted textual documents and images from a unified position, by means of their partial randomization and metered adding of uncorrelated Gaussian noise. The changes occurring in a text or image under progressive disordering have been studied for the first time via analyzing a gradual evolution of their LZA-based compression coefficients S during the randomization or noising processes. The S parameter has been measured for the randomized natural-language texts and for the noisy or randomized images. We have demonstrated for the first time that the dependences $S(N)$ for both sample text and image behave in the same manner: first we observe the upper plateau $S \approx const$ at relatively small randomization parameters N, where the structural order and the information contents of a text or image persist (see also **Fig. 1**); then the $S(N)$ function changes abruptly in the intermediate region

given approximately by the equality $N \approx L$ (or $N \approx M$) and, finally it attains the lower plateau $S \approx const$ at $N \gg L$ ($N \gg M$), which corresponds to a completely disordered and so uninformative text or image.

For the text, the sentence-level randomization almost does not change the compressibility, whereas the word-level and, especially, the character-level types of randomization lead to a pronounced decrease in the S parameter. In all cases, the strongest changes occur in the vicinity of $N \approx L$. This shows that the typical number N of randomization cycles needed to destroy the order completely in a symbolic sequence is comparable with the total number L of its elements. For the image represented as a one-dimensional sequence of red-channel pixel values, the compression coefficient decreases significantly when the number N of pixel-randomization cycles reaches the total number M of pixels ($N \approx M$), quite similar to the case of our randomized text ($N \approx L$). Under Gaussian noise, the main decrease in S occurs at the σ values that correspond approximately to the standard deviation σ_0 of pixel intensities in the original image.

Unlike the case of texts, the limiting compression values for the strongly noised or randomized images remain essentially nonzero. This is explained mainly by a limited set of possible intensity values and the additional restrictions imposed on the pixel-value distribution by the image histogram. The additional JPEG-size analysis has led to qualitatively similar conclusions about the influence of randomization and noise on file compressibility. Finally, the dependences $S(N)$ and $K(N)$ for the image are qualitatively different from $S(\sigma)$ and $K(\sigma)$: the former reveal a clear initial plateau, contrary to the latter. This emphasizes different effects of randomization and noising on the structural order in the image. However, the dependences *PSNR*(N) and *PSNR*(σ) are much more similar to each other. Finally, it is worthwhile that the *SE*(σ) dependence also reveals the initial plateau at the smallest σ, unlike the *PSNR*(σ) dependence. In a sense, the S and K parameters are closer to *SE* than to *PSNR* in this respect.

The compression measure applied to images should be distinguished from purely histogram-based characteristics. In principle, a corrupted image can have exactly the same intensity histogram as the original image, while its spatial organization can be at least partly lost. This holds true, e.g., for an original image and its randomized versions. One can see that their compressibilities are radically different (see **Fig. 5a**), i.e. the compression parameter can detect the loss of spatial organization which is invisible in the histogram alone (see Fig. 7a). In other words, our LZA-based compression parameter for images has an important advantage over histogram-based parameters. Note in this respect that the computational complexity of compressibility is higher than that of *SE* but lower than that of *PSNR*.

Of course, modeling of stochastically perturbed textual and graphical data by randomization and Gaussian noising is restricted enough. The other factors hindering us from too categorical conclusions are a limited set of experimental samples and prematurity of any S-based detection-accuracy estimates. Nonetheless, our experimental dependences $S(N)$ and $S(\sigma)$ testify that the compression parameters can be, in principle, used for quantifying the loss of structural order in the data and detecting noised natural-language texts or images.

We perceive that further promising work can be related to broadening of the experimental objects under test, comparing different compression algorithms, as well as different methods of image unfolding and combinations of compression parameters with other descriptors of order and noise. This would aid in formulating the practical criteria of noise detection based upon compressibility.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [O.Ku.]; methodology, [O.Kr.]; validation, [O.Kr.]; formal analysis, [O.Kr., O.Ku.]; investigation, [O.Kr.]; resources, [O.Kr., O.Ku.]; data curation, [O.Kr.]; writing – original draft preparation, [O.Kr.]; writing – review and editing, [O.Ku.]; visualization, [O.Kr., O.Ku.]; supervision, [O.Ku.]; project administration, [O.Ku.].

All the authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Dębowski, Ł. (2011). Excess entropy in natural language: present state and perspectives. *Chaos*, *21*, 037105. https://doi.org/10.1063/1.3630929

[2] Amancio, D. R., Altmann, E. G., Rybski, D., Oliveira Jr, O. N., & da F Costa, L. (2013). Probing the statistical properties: application to the Voynich manuscript. *PLoS ONE*, *8*, e67310. https://doi.org/10.1371/journal.pone.0067310

[3] Dębowski, Ł. (2015). The relaxed Hilberg conjecture: a review and new experimental support. *J. Quant. Linguistics*, *22*(4), 311–337. https://doi.org/10.1080/09296174.2015.1106268

[4] Timm, T., & Schinner, A. (2020). A possible generating algorithm of the Voynich manuscript. *Cryptologia*, *44*, 1–19. https://doi.org/10.1080/01611194.2019.1596999

[5] Feng, J., & Yu S. (2024). A method for measuring word sequence complexity of text. *J. Quant. Linguistics*, *32*(1), 82–102. https://doi.org/10.1080/09296174.2024.2417448

[6] Wieczyński, P., & Dębowski, Ł. (2025). Long-range dependence in word time series: the cosine correlation of embeddings. *Entropy*, *27*, 613. https://doi.org/10.3390/e27060613

[7] Cover, T., & Thomas, J. (1991). *Elements of Information Theory*. Wiley. 542 p.

[8] Akagi, T., Funakoshi, M., & Inenaga, S. (2023). Sensitivity of string compressors and repetitiveness measures. *Information and Computation*, 291, 104999. https://doi.org/10.1016/j.ic.2022.104999

[9] Jiang, Z., Yang, M., Tsirlin, M., Tang, R., Dai, Y., & Lin, J. (2023). “Low-resource” text classification: a parameter-free classification method with compressors. *Findings of the Association for Computational Linguistics: ACL 2023*, 6810–6828. https://doi.org/10.18653/v1/2023.findings-acl.426

[10] Karjus, A., Canet Solà, M., Ohm, T., Ahnert, S. E., & Schich, M. (2023). Compression ensembles quantify aesthetic complexity and the evolution of visual art. *EPJ Data Science*, *12*, 21. https://doi.org/10.1140/epjds/s13688-023-00397-3

[11] Merhav, N. (2026). Lempel–Ziv complexity, empirical entropies, and chain rules. *IEEE Trans. Inform. Theory*, *72*(5), 3486–3496. https://doi.org/10.1109/TIT.2026.3668013

[12] Lande, D. V., & Snarskii, A. A. (2007). On the role of autocorrelations in texts. *arXiv:0710.0225*. https://doi.org/10.48550/arXiv.0710.0225
[13] Gurzadyan, A. V., & Allahverdyan, A. E. (2016). Non-random structures in universal compression and the Fermi paradox. *Eur. Phys. J. Plus*, *131*, 26. https://doi.org/10.1140/epjp/i2016-16026-6
[14] Deng, W., Xie, R., Deng, S., & Allahverdyan, A. E. (2021). Two halves of a meaningful text are statistically different. *J. Statist. Mechanics: Theory and Experiment*, *3*, 033413. https://doi.org/10.1088/1742-5468/abe947
[15] Serrano, M. Á., Flammini, A., & Menczer, F. (2009). Modeling statistical properties of written text. *PLoS ONE*, 4(4), e5372. https://doi.org/10.1371/journal.pone.0005372
[16] Gonzalez, R., & Woods, R. (2018). *Digital Image Processing*. Pearson. 1020 p.
[17] Tang, C., Yang, X., & Zhai, G. (2015). Noise estimation of natural images via statistical analysis and noise injection. *IEEE Trans. on Circuits and Systems for Video Technol.*, *25*(8), 1283–1294 https://doi.org/10.1109/TCSVT.2014.2380196
[18] Zhang, Z. (2015). Image noise: detection, measurement, and removal techniques. In *Knoxville: Department of Electr. Eng. & Comp. Sci.* (10 pp.). https://zzutk.github.io/docs/reports/2015.07%20-%20Image%20Noise%20Detection,%20Measurement,%20and%20Removal%20Techniques.pdf
[19] Wu, Z., Lu, X., & Deng, Y. (2015). Image edge detection based on local dimension: a complex networks approach. *Physica A*, *440*, 9–18. https://doi.org/10.1016/j.physa.2015.07.020
[20] Dovhan, I. Ya., Kushnir, O. S., Furhala, Yu. M., & Panochko, H. I. (2022). Analogies between images and texts: burstiness phenomenon in texts and digital images. *Electronics and Information Technologies*, *17*, 3–15. https://doi.org/10.30970/eli.17.1
[21] Kushnir, O., Kravchuk, O., Drebot, A., & Franiv, V. (2025). Properties of networks built on digital images. *Electronics and Information Technologies*, *28*, 1–15. https://doi.org/10.30970/eli.29.2

КОЕФІЦІЄНТ СТИСНЕННЯ В АНАЛІЗІ ТЕКСТІВ І ЗОБРАЖЕНЬ ІЗ ШУМАМИ

Олексій Кравчук, Олег Кушнір
Кафедра оптоелектроніки та інформаційних технологій,
Львівський національний університет імені Івана Франка,
вул. Ген. Тарнавського, 107, м. Львів, 79017, УКРАЇНА

АНОТАЦІЯ

Вступ. Проаналізовано коефіцієнт стиснення S, пов'язаний з двома версіями алгоритму Лемпеля–Зіва (АЛЗ, англ. LZA), як кількісну характеристику структурного порядку в зашумлених або пошкоджених даних на прикладі тексту природною мовою та цифрового зображення. Для моделювання частково пошкоджених тексту або зображення ми рандомізували лінгвістичні елементи першого та додавали некорельований ґаусівський шум до другого або рандомізували його пікселі.

Матеріали та методи. Художній текст «Мобі Дік» Г. Мелвілла був рандомізований на лінгвістичних рівнях речень, слів і символів. Кількість циклів рандомізації N змінювали в діапазоні від 0 до 10^9. Проаналізовано також червоний канал 8-бітного зображення «Peppers» із розмірами $M = 512\times512$, до якого було додано ґаусівський шум зі стандартними відхиленнями σ у діапазоні від 0 до 400. Загалом було згенеровано 50 рандомізованих і 50 зашумлених версій зображення. Для застосування стиснення АЛЗ зображення розгортали в одновимірну послідовність інтенсивностей пікселів і обробляли відповідне текстове представлення зображення як звичайний текст.

Результати та обговорення. Рандомізація на рівні речень майже не змінює значення S, оскільки вона не змінює внутрішньої будови слів і речень, а АЛЗ має локальний характер. Рандомізація на рівні слів помірно зменшує S, а рандомізація на рівні символів приводить до найбільших змін S і майже нестисливих текстових файлів за найбільших N. Результати, одержані для двох версій АЛЗ, слабко відрізняються за впливом рандомізації речень на стисливість. Для рандомізованих зображень значення S різко зменшується, коли N наближається до загальної кількості пікселів M, а потім сягає насичення. Найбільші зміни S для зображень із шумами маємо в області $\sigma \in [10; 100]$.

Висновки. Метрики на основі стиснення АЛЗ чутливі до втрати локального порядку в будь-яких даних із шумами – і в рандомізованих текстах, і в рандомізованих та зашумлених зображеннях. Оскільки АЛЗ базується на повтореннях і локальних закономірностях у послідовності символів, зменшення S вказує на часткове руйнування порядку, спричинене рандомізацією або шумом. Отже, коефіцієнт стиснення може бути мірилом невпорядкованості в послідовностях лінгвістичних символів і в мережах пікселів, і, отже, може допомогти у виявленні зашумлених даних.

Ключові слова: стиснення даних, аналіз даних із шумами, розпізнавання шумів, рандомізація, опрацювання природної мови, опрацювання зображень.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
06 August, 2026	23 September, 2026	25 September, 2026	28 September , 2026

Electronics and Information Technologies, 2026, 35, 45–56
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.93

SEGMENTATION OF COMPLEX STRUCTURES IN POINT CLOUDS USING DEEP NEURAL NETWORKS

Viktor Vdovychenko* & ***Yuriy Furgala***
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv,
50 Drahomanova Street, Lviv, 79005, UKRAINE

Vdovychenko V. M., & Furgala Y. M. (2026). Segmentation of Complex Structures in Point Clouds Using Deep Neural Networks. *Electronics and Information Technologies*, 35, 45–56. https://doi.org/10.30970/eli.35.4

ABSTRACT

Background. Point cloud part segmentation assigns a semantic part label to every point and provides a more detailed description of object geometry than object-level classification. Architecture selection remains difficult because models with similar aggregate benchmark scores may differ substantially in category-level reliability and computational cost.

Materials and Methods. This study compares PointNet++, Dynamic Graph Convolutional Neural Network (DGCNN), and Point Transformer on the ShapeNet Part benchmark, which represents sixteen object categories with fifty part labels. Final evaluation was performed on 2,874 test shapes represented by 2,048 points. Segmentation quality was assessed using point accuracy, class-average accuracy, class mean intersection over union (mIoU), and instance mean intersection over union (mIoU). A separate batch-size-one benchmark measured inference latency, throughput, and graphics memory use.

Results and Discussion. Point Transformer achieved the highest overall quality, with 94.4% point accuracy, 87.7% class-average accuracy, 82.9% class mIoU, and 85.7% instance mIoU. DGCNN reached 85.3% instance mIoU while requiring only 15.87 ms per shape, compared with 53.39 ms for Point Transformer and 344.62 ms for PointNet++. Aggregate differences were small, but category-level results were more pronounced. Point Transformer produced the highest mIoU in eight of sixteen categories, while DGCNN and PointNet++ led in five and three categories, respectively. The largest architecture-dependent differences were observed for Motorbike, Earphone, and Rocket, demonstrating that similar overall scores can conceal distinct failure patterns.

Conclusion. No architecture dominated all practical criteria. Point Transformer provided the strongest segmentation quality, whereas DGCNN offered the most favorable measured speed-quality trade-off. PointNet++ remained competitive in quality but showed substantially higher latency in the evaluated implementation. Model selection should therefore consider category-level behavior, input configuration, and computational constraints together with aggregate mIoU.

Keywords: point cloud segmentation, part segmentation, ShapeNet Part, PointNet++, DGCNN, Point Transformer

INTRODUCTION

Point clouds are a fundamental representation of three-dimensional geometry and are widely used in robotics, autonomous systems, industrial inspection, digital twins, reconstruction, and geometric analysis [1]. Unlike raster images, point clouds are unordered and irregularly sampled, and they do not provide a fixed neighborhood structure.

© 2026 Viktor Vdovychenko & Yuriy Furgala. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

These properties make direct processing by conventional grid-based convolution difficult and motivate neural architectures designed specifically for point sets.

Object-level classification assigns a single semantic label to an entire point cloud, whereas part segmentation assigns a label to every point. The latter therefore provides a substantially more detailed representation of an object's internal structure and is directly relevant to applications that require localization of functional or geometric parts, extraction of dimensions, or subsequent estimation of physical characteristics. PointNet demonstrated that point sets can be processed directly with permutation-invariant operations and established a common foundation for both classification and segmentation [2].

Modern point-cloud networks use different mechanisms to capture local and global geometric relationships. PointNet++ introduces hierarchical sampling, neighborhood grouping, and local PointNet operations to learn features at multiple spatial scales [3]. DGCNN represents points as a dynamically reconstructed graph and applies EdgeConv to learn relationships between each point and its local neighbors in feature space [4]. Point Transformer adapts self-attention to point sets and incorporates relative position information into feature aggregation [5]. These architectural differences can influence not only segmentation quality but also computational cost and sensitivity to individual object categories.

Recent research continues to expand transformer-based and segmentation-oriented point-cloud processing. Point Transformer V3 emphasizes scalable and efficient serialized processing [8], RangeViT applies transformer representations to large-scale LiDAR semantic segmentation [9], and CamPoint explores visibility-based local and global context modeling for point cloud segmentation [10]. Despite this progress, classic architectures such as PointNet++, DGCNN, and the original Point Transformer remain useful reference points because they represent three distinct and widely used design principles: hierarchical point aggregation, dynamic graph learning, and position-aware attention.

Direct comparison of results reported in independent studies is difficult because implementations commonly differ in the number of sampled points, input features, data augmentation, optimizer settings, voting strategies, checkpoint selection, and hardware platforms. Small differences in aggregate mean Intersection over Union (mIoU) may therefore be difficult to interpret without a controlled benchmark. Moreover, an architecture that has the best average score may not be preferable when latency, model size, or category-level reliability are important deployment constraints.

The aim of this study is to compare PointNet++, DGCNN, and Point Transformer for object part segmentation on the ShapeNet Part benchmark and to analyze their segmentation quality, training dynamics, category-level behavior, and computational efficiency. The comparison uses a common final test protocol with 2,048 points per shape, 2,874 test objects, batch size one, and the same graphics processing unit. Because the reference implementations do not expose identical input pipelines, the comparison also explicitly reports implementation-specific differences. The study is intended to identify practical trade-offs that are not visible from a single aggregate mIoU value.

MATERIALS AND METHODS

Dataset

Experiments were conducted on ShapeNet Part, a benchmark derived from the ShapeNet collection and designed for fine-grained part-level annotation of three-dimensional objects [6, 7]. The complete benchmark contains 16,881 shapes from 16 object categories with 50 semantic part labels in total. Each object category contains its own subset of valid part labels. The categories evaluated in this study were Airplane, Bag, Cap, Car, Chair, Earphone, Guitar, Knife, Lamp, Laptop, Motorbike, Mug, Pistol, Rocket, Skateboard, and Table. Representative examples of ShapeNet Part objects with ground-truth part annotations are shown in **Fig. 1**.

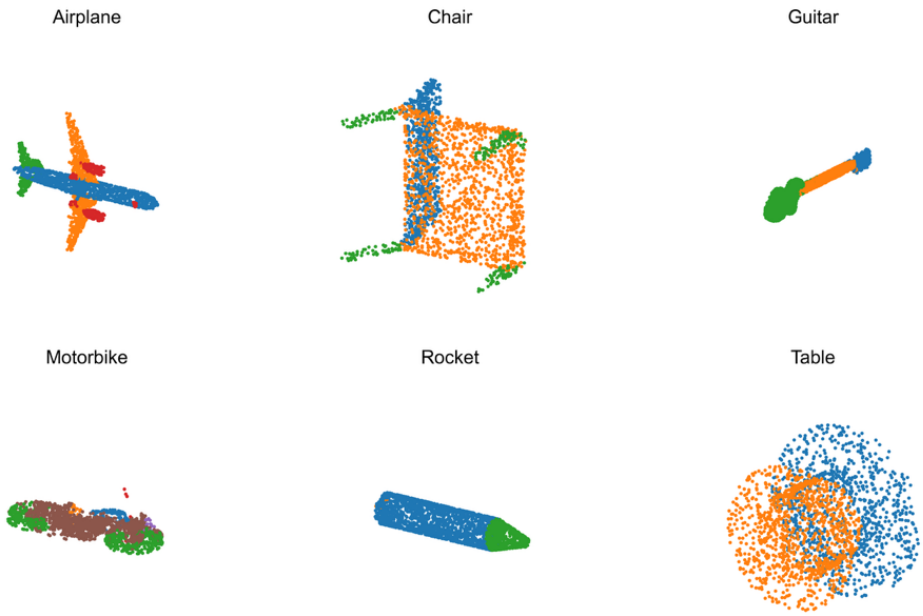


Fig. 1. Representative ShapeNet Part objects with ground-truth part segmentation

The final benchmark used 2,874 test shapes, and every shape was represented by 2,048 points. The evaluated implementations used the standard ShapeNet Part training/test organization provided by their respective repositories, while the final quantitative comparison reported in this paper was recomputed on the common 2,874-object test set. This distinction is important because repository-specific preprocessing and training pipelines may not be identical even when the same benchmark is used.

Evaluated Architectures

PointNet++. PointNet++ extends PointNet by constructing a hierarchical representation of a point cloud. Farthest point sampling selects representative centroids, local neighborhoods are grouped around these points, and PointNet-style functions encode increasingly abstract features [3].

DGCNN. Dynamic Graph Convolutional Neural Network constructs a k-nearest-neighbor graph and applies EdgeConv to learn local relationships between a point and neighboring features [4]. The graph is recomputed in feature space across layers, allowing local connectivity to adapt as representations evolve. The evaluated part-segmentation configuration used k = 40, embedding dimension 1,024, 2,048 points.

Point Transformer. Point Transformer applies position-aware self-attention to point clouds. Relative spatial information is incorporated into attention-based feature aggregation so that the network can model both geometric and feature relationships [5].

Training and Evaluation Protocol

All three models were trained from scratch using their reference part-segmentation implementations. PointNet++ was trained for 250 epochs with batch size 16 and the Adam optimizer. DGCNN was trained for 250 epochs with batch size 8, stochastic gradient descent, momentum 0.9, a cosine learning-rate schedule, and dropout 0.5. Point Transformer was trained for 250 epochs using its SGD configuration. The best checkpoints were selected during training according to the validation criteria implemented in the corresponding repositories.

The experiments were performed using publicly available PyTorch implementations of the three architectures. For reproducibility, the exact source repositories used in this study were: PointNet++ [11]; DGCNN [12]; and Point Transformer (unofficial implementation) [13]. The reported inference latency and throughput therefore characterize these specific implementations and software revisions rather than architecture-independent theoretical performance.

For the final benchmark, every model processed 2,048 points per object with batch size one. Twenty warm-up batches were excluded before latency measurements. The benchmark evaluated all 2,874 test shapes and reported point accuracy, class-average accuracy, class-average mIoU, instance-average mIoU, latency, throughput, memory consumption, number of parameters, and checkpoint size. PointNet++ was evaluated with one vote to avoid introducing test-time ensembling into the comparison.

The final inference protocol was standardized where possible. Therefore, the results should be interpreted as a comparison of the evaluated implementations rather than as an ablation study that isolates architectural differences while keeping all input features and preprocessing steps identical. The main training and evaluation settings for the three models are summarized in **Table 1**.

Table 1. **Experimental configuration of the evaluated segmentation models**

Parameter	PointNet++	DGCNN	Point Transformer
Input points	2048	2048	2048
Training epochs	250	250	250
Training batch size	16	8	16
Optimizer	Adam	SGD	SGD
Final test batch size	1	1	1
Test shapes	2874	2874	2874

Evaluation Metrics

Point accuracy was computed as the proportion of point labels predicted correctly over all points in the test set. Class-average accuracy gives equal importance to the semantic part classes and is therefore more sensitive to less frequent labels than overall point accuracy.

Intersection over Union (IoU) was used as the principal segmentation measure. For each part, IoU quantifies the overlap between the predicted and ground-truth point sets relative to their union. Class mIoU averages segmentation quality across the 16 object categories, whereas instance mIoU averages the part IoU over individual test shapes. Reporting both metrics is important because category frequencies are not uniform and strong performance on common categories can otherwise dominate an instance-weighted score.

Category-level mIoU was additionally analyzed for all 16 ShapeNet Part categories. This analysis was used to identify cases in which similar aggregate scores conceal architecture-specific strengths or weaknesses. For selected categories discussed in detail, variability across individual test shapes was additionally characterized by the standard deviation of per-shape IoU and is reported as ± SD.

Computational Efficiency Evaluation

Computational efficiency was measured using a separate inference benchmark with batch size one. Mean latency was expressed in milliseconds per shape, and throughput

was calculated as processed shapes per second. Graphics processing unit memory was reported as maximum allocated and reserved memory during the benchmark.

Latency measurements excluded data loading and result saving and were collected after 20 warm-up batches. The use of batch size one reflects a low-latency inference scenario and makes the measurements directly comparable across the three benchmark scripts.

Experimental Environment

The experiments were conducted on a workstation equipped with an NVIDIA GeForce GTX 1080 Ti graphics processing unit with 11 GB of video memory, an Intel Core i5-10600K processor, and 32 GB of random-access memory. The segmentation experiments were executed under Ubuntu 24.04 using Python-based PyTorch implementations [14]. Separate software environments were used where required by repository-specific dependencies, while the final inference benchmark used the same graphics processing unit for all models.

RESULTS AND DISCUSSION

Overall Segmentation Performance

Table 2 summarizes the final segmentation quality obtained on the common ShapeNet Part test set. All three architectures achieved point accuracy above 94% and instance mIoU above 85%, indicating that the overall performance differences are relatively small. Point Transformer achieved the strongest aggregate result, with 94.4% point accuracy, 87.7% class-average accuracy, 82.9% class mIoU, and 85.7% instance mIoU. DGCNN obtained 94.3% point accuracy and 85.3% instance mIoU, while PointNet++ achieved 94.2% and 85.1%, respectively.

Table 2. **Overall segmentation performance on the ShapeNet Part test set**

Model	Point accuracy, %	Class avg. accuracy, %	Class mIoU, %	Instance mIoU, %
PointNet++	94.2	87.2	81.8	85.1
DGCNN	94.3	84.2	81.4	85.3
Point Transformer	94.4	87.7	82.9	85.7

The ranking differs slightly depending on the selected metric. PointNet++ produced higher class-average accuracy than DGCNN, whereas DGCNN produced higher instance mIoU. The lower class-average accuracy of DGCNN compared with PointNet++ and Point Transformer suggests that its errors are distributed less uniformly across semantic part classes. Because class-average accuracy gives equal weight to each part class, weaker performance on less frequent or more difficult part labels can reduce this metric even when instance-level mIoU remains competitive. This indicates that a single aggregate measure is not sufficient to characterize segmentation behavior. The maximum difference in instance mIoU between the three models was only 0.6 percentage points, but the class mIoU difference between DGCNN and Point Transformer reached approximately 1.5 percentage points. The category-level analysis below shows that these apparently small aggregate differences are composed of much larger gains and losses on individual object types.

Training Dynamics

Fig. 2 compares test accuracy and instance mIoU over the training trajectories. All three networks improved rapidly during the initial epochs and then entered a slower

refinement stage. PointNet++ reached strong segmentation quality comparatively early, and its best checkpoint used in the benchmark was obtained at epoch 136. Point Transformer continued improving over a longer interval, with the benchmark checkpoint selected at epoch 170. DGCNN showed stronger epoch-to-epoch fluctuations in the test curves, including several temporary drops, but recovered and converged to an instance mIoU comparable with the other architectures.

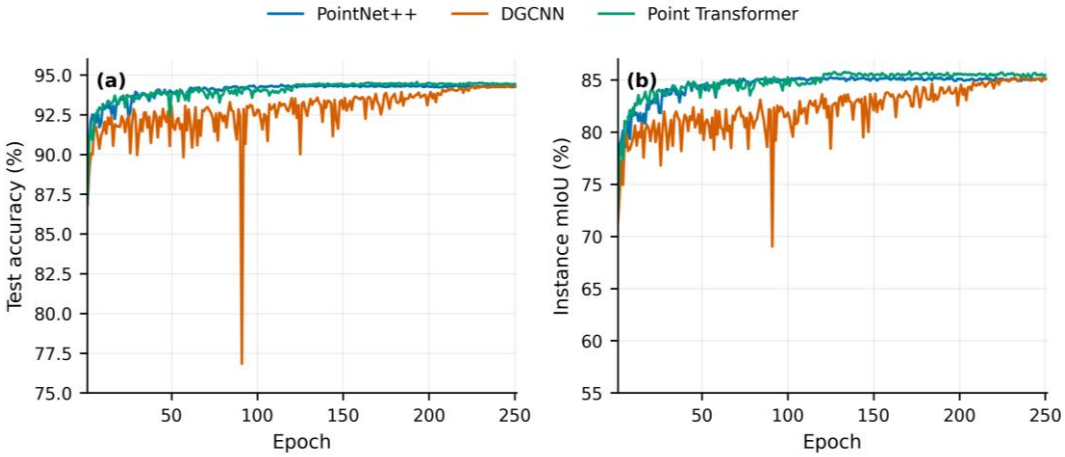


Fig. 2. Training dynamics of the evaluated segmentation models: (a) test accuracy and (b) instance mIoU

The training curves should be interpreted as optimization behavior rather than as a direct ranking of generalization quality. The final benchmark was recomputed from saved checkpoints under a separate batch-size-one protocol; therefore, the values in **Table 2** are the primary basis for the final comparison. Nevertheless, the curves demonstrate that architectures with similar final mIoU can reach that level through different convergence trajectories.

Per-Category Segmentation Performance

Fig. 3 presents mIoU for each of the 16 ShapeNet Part categories. Point Transformer achieved the highest category mIoU in eight categories, DGCNN in five, and PointNet++ in three. The results therefore support the aggregate ranking of Point Transformer while also showing that no model dominates every object type.

The largest architecture-dependent difference was observed for Motorbike. DGCNN reached 61.3 ± 10.1% mIoU, PointNet++ 71.2 ± 9.8%, and Point Transformer 73.5 ± 10.0%, giving a spread of more than 12 percentage points. Earphone also revealed a clear weakness of PointNet++, which achieved 69.9 ± 17.3% compared with 77.6 ± 18.8% for DGCNN and 78.6 ± 17.2% for Point Transformer. The relatively large standard deviations indicate substantial variation in segmentation quality across individual objects within these categories.

Rocket was the most difficult category for all three models. DGCNN achieved 55.1 ± 11.7%, Point Transformer 58.2 ± 11.2%, and PointNet++ 60.9 ± 10.4%, making PointNet++ the strongest model on this category despite its lower aggregate score. In contrast, Laptop was segmented reliably by every architecture, with mIoU between 95.1% and 95.9%. DGCNN produced the best Laptop result and also led on Airplane, Guitar, Knife, and Lamp. PointNet++ led on Pistol, Rocket, and Skateboard. Point Transformer performed best on Bag, Cap, Car, Chair, Earphone, Motorbike, Mug, and Table.

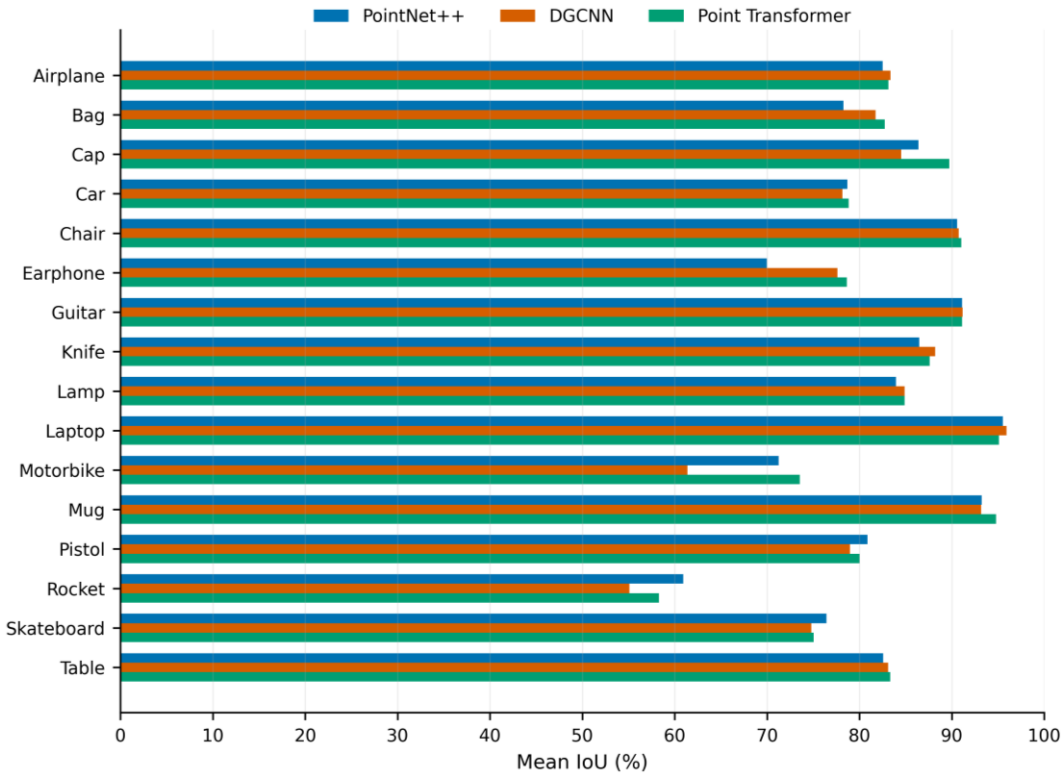


Fig. 3. Per-category mIoU comparison for PointNet++, DGCNN, and Point Transformer on ShapeNet Part

Overall, the per-category comparison demonstrates that aggregate mIoU should be supplemented by category-level analysis and measures of variability across individual objects when segmentation reliability for a specific object family matters.

Computational Efficiency and Quality–Efficiency Trade-off

The three networks differed much more strongly in inference efficiency than in segmentation quality. **Table 3** summarizes the batch-size-one benchmark. DGCNN was the fastest model, requiring 15.87 ms per shape and processing approximately 62.99 shapes/s. Point Transformer required 53.39 ms per shape and reached 18.73 shapes/s. PointNet++ was substantially slower in the evaluated implementation, with a mean latency of 344.62 ms per shape and throughput of 2.90 shapes/s.

Table 3. **Measured computational efficiency in the batch-size-one benchmark**

Model	Latency ms	Rate shapes/s	GPU MB
PointNet++	344.62	2.90	132.9
DGCNN	15.87	62.99	157.7
Point Transformer	53.39	18.73	99.1

Point Transformer showed the lowest maximum allocated graphics memory during the measured batch-size-one inference, requiring 99.1 MB, compared with 132.9 MB for PointNet++ and 157.7 MB for DGCNN.

Fig. 4 makes the practical trade-off especially clear. Moving from DGCNN to Point Transformer increases instance mIoU by approximately 0.4 percentage points, but increases mean latency by a factor of about 3.4. PointNet++ does not provide an aggregate

quality advantage over the other two models in this benchmark and is approximately 6.5 times slower than Point Transformer and 21.7 times slower than DGCNN. These values characterize the tested implementations and should not be interpreted as architecture-independent theoretical speed ratios; nevertheless, they are directly relevant to deployment when the same software and hardware stack is used.

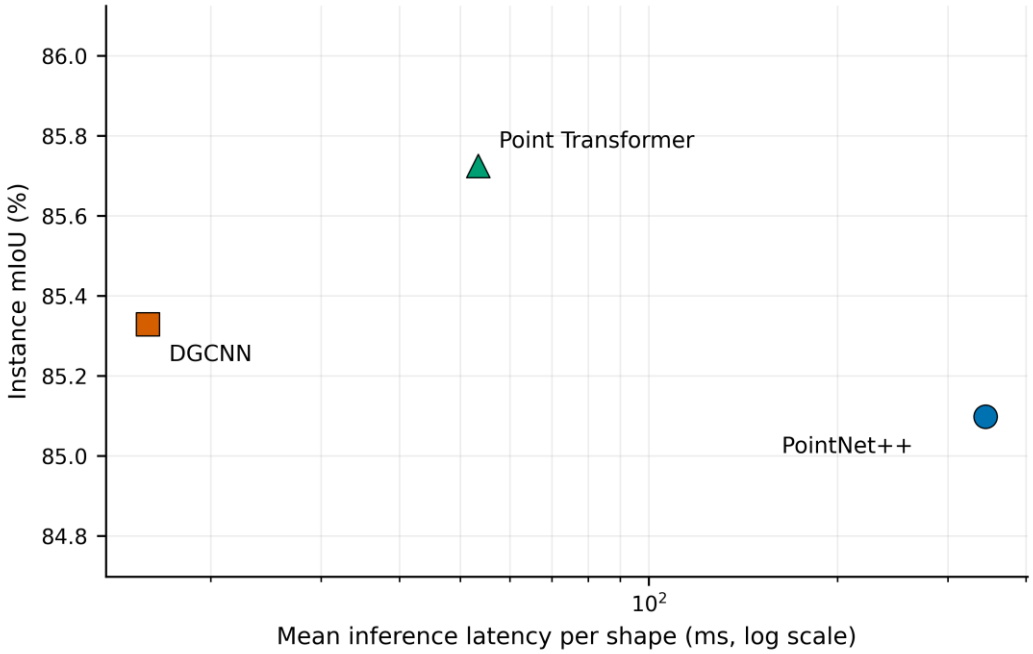


Fig. 4. Quality–efficiency trade-off between instance mIoU and mean inference latency. The latency axis is logarithmic.

Practical Trade-offs and Study Limitations

The experimental results demonstrate that model selection depends on the intended application. Point Transformer is preferable when the primary objective is maximum segmentation quality and a moderate increase in inference cost is acceptable. DGCNN provides the strongest measured speed-quality compromise: its instance mIoU is only 0.40 percentage points below Point Transformer while its inference is substantially faster. PointNet++ remains a competitive segmentation baseline and performs best on several categories, including Rocket, but its measured latency makes it less attractive for low-latency deployment in the evaluated implementation.

Several limitations should be considered when interpreting the results. First, the experiments use a single synthetic/CAD-derived benchmark and therefore do not quantify robustness to sensor noise, occlusion, missing points, or non-uniform sampling in real-world scans. Second, only three architectures were evaluated, so the results do not represent the full range of recent point-cloud segmentation methods. Third, all efficiency measurements were obtained on one hardware platform and are influenced by repository-specific kernels and implementation details.

Finally, ShapeNet Part evaluates object-level part segmentation rather than semantic segmentation of large scenes. The conclusions are therefore most directly applicable to normalized object point clouds. Extending the comparison to real sensor data, larger scenes, and downstream geometric measurements will be necessary to determine how the observed trade-offs transfer to practical three-dimensional perception systems.

CONCLUSION

This study compared PointNet++, DGCNN, and Point Transformer for part segmentation of point clouds on the ShapeNet Part benchmark. The final evaluation used 2,874 test shapes, 2,048 points per object, and a common batch-size-one inference protocol. Segmentation performance was analyzed using point accuracy, class-average accuracy, class mIoU, instance mIoU, per-category mIoU, training dynamics, and computational efficiency.

Point Transformer achieved the highest aggregate segmentation quality, reaching 94.4% point accuracy, 82.9% class mIoU, and 85.7% instance mIoU. DGCNN and PointNet++ remained close in overall quality, with instance mIoU values of 85.3% and 85.1%, respectively. The small aggregate differences, however, concealed substantially larger category-specific variation. Point Transformer produced the highest mIoU in eight categories, DGCNN in five, and PointNet++ in three. Motorbike, Earphone, and Rocket showed the most informative architecture-dependent differences, while Laptop was consistently segmented with high accuracy by all models.

The computational benchmark revealed a much stronger separation between the architectures. DGCNN achieved the lowest mean latency of 15.87 ms per shape and the highest throughput of 62.99 shapes/s. Point Transformer required 53.39 ms per shape, whereas PointNet++ required 344.62 ms. These results show that a model with the highest segmentation quality is not automatically the most suitable model for a latency-constrained system.

Overall, Point Transformer provides the strongest segmentation quality in the evaluated setup, while DGCNN offers the most favorable measured speed-quality trade-off. PointNet++ remains useful as a hierarchical baseline and demonstrates strengths on selected categories, but its inference cost was substantially higher in the tested implementation. The results confirm that architecture selection should combine aggregate mIoU with category-level reliability and deployment-oriented efficiency metrics.

Future work will focus on methods for determining geometric and physical object parameters from three-dimensional point cloud data. The study will consider point clouds containing isolated objects of different simple geometric shapes and will investigate both algorithmic and neural-network-based approaches. Classification models will be used to identify the geometric type of an object, after which geometry-specific algorithms will estimate its relevant dimensions, shape characteristics, and derived physical parameters. The proposed framework will enable a systematic comparison of analytical and learning-based approaches and will investigate whether combining object recognition with geometry-aware parameter estimation can improve the reliability and general applicability of measurements obtained directly from point clouds.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author received no financial support for the research, writing, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [V.V., Y.F.]; methodology, [V.V.]; validation, [V.V.; formal analysis, [V.V.]; investigation, [V.V.]; resources, [V.V.]; data curation, [V.V.]; writing – original draft

preparation, [V.V.]; writing – review and editing, [V.V.]; visualization, [V.V.]; supervision, [V.V., Y.F.]; project administration, [V.V., Y.F.].

All authors have read and agreed to the published version of the manuscript.

DECLARATION ON GENERATIVE AI

During the preparation of this manuscript, the authors used ChatGPT for language editing, improving the structure and flow of author-provided text, and assistance with formatting according to the journal template. The tool was not used to generate experimental data, numerical benchmark results, citations, or scientific conclusions. After using ChatGPT, the authors reviewed and edited the resulting text and take full responsibility for the content of the publication.

REFERENCES

[1] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2021). Deep learning for 3D point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12), 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434

[2] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 652–660. https://doi.org/10.1109/CVPR.2017.16

[3] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.02413

[4] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics, 38(5), 1–12. https://doi.org/10.1145/3326362

[5] Zhao, H., Jiang, L., Jia, J., Torr, P. H. S., & Koltun, V. (2021). Point Transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision, 16259–16268. https://doi.org/10.1109/ICCV48922.2021.01595

[6] Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., & Guibas, L. (2016). A scalable active framework for region annotation in 3D shape collections. ACM Transactions on Graphics, 35(6), 1–12. https://doi.org/10.1145/2980179.2980238

[7] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., & Yu, F. (2015). ShapeNet: An information-rich 3D model repository. arXiv. https://doi.org/10.48550/arXiv.1512.03012

[8] Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2024). Point Transformer V3: Simpler, faster, stronger. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4840–4851. https://doi.org/10.1109/CVPR52733.2024.00463

[9] Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., & Marlet, R. (2023). RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5240–5250. https://doi.org/10.1109/CVPR52729.2023.00507

[10] Zhang, J., Luo, Y., Zhang, Z., Nie, X., & Li, B. (2025). CamPoint: Boosting point cloud segmentation with virtual camera. Proceedings of the IEEE/CVF Conference

on Computer Vision and Pattern Recognition, 11822–11832. https://doi.org/10.1109/CVPR52734.2025.01104
[11] Yan, X. (2019). *Pointnet_Pointnet2_pytorch: PyTorch implementation of PointNet and PointNet++* [Source code]. GitHub. https://github.com/yanx27/Pointnet_Pointnet2_pytorch
[12] Tao, A. (n.d.). *dgcnn.pytorch: A PyTorch implementation of Dynamic Graph CNN for Learning on Point Clouds* [Source code]. GitHub. https://github.com/antao97/dgcnn.pytorch
[13] Ji, J. (n.d.). *Point-Transformer: A PyTorch implementation of Point Transformer for batch-mode point cloud processing* [Source code]. GitHub. https://github.com/ji-code25/Point-Transformer
[14] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32. https://doi.org/10.48550/arXiv.1912.01703

СЕГМЕНТАЦІЯ СКЛАДНИХ СТРУКТУР У ХМАРАХ ТОЧОК ЗА ДОПОМОГОЮ ГЛИБИННИХ НЕЙРОННИХ МЕРЕЖ

Віктор Вдовиченко*, Юрій Фургала
Факультет електроніки та комп'ютерних технологій,
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, Львів, 79005, УКРАЇНА

АНОТАЦІЯ

Вступ. Сегментація частин хмар точок призначає семантичну мітку кожній точці та забезпечує детальніший опис геометрії об'єкта, ніж класифікація цілого об'єкта. Вибір архітектури ускладнюється тим, що моделі з близькими інтегральними показниками можуть істотно відрізнятися в окремих категоріях і за обчислювальною ефективністю.

Матеріали та методи. Для порівняльного оцінювання обрано три репрезентативні архітектури обробки хмар точок, що відображають різні підходи до вилучення геометричних ознак: PointNet++ (ієрархічна агрегація локальних ознак), DGCNN (динамічна побудова графа сусідів і графові згортки) та Point Transformer (механізм самоуваги для неструктурованих даних). Порівняння проведено на наборі ShapeNet Part (16 категорій об'єктів, 50 міток частин).Фінальне оцінювання виконано на 2874 тестових об'єктах, представлених 2048 точками. Якість визначали за точністю точок, середньою точністю за класами, середнім коефіцієнтом Жаккара (КЖ) за категоріями та середнім КЖ за об’єктами. Для вимірювання затримки виведення, пропускної здатності та обсягу використаної відеопам'яті було проведено окремий тест із розміром пакета 1.

Результати. Point Transformer показав найкращу загальну якість: точність 94,4%, середня точність за класами 87,7%, середній КЖ за категоріями 82,9% та середній КЖ за об'єктами 85,7%. При цьому DGCNN досягла близького показника середнього КЖ за об'єктами (85,3%), але значно швидше: 15,87 мс на один об'єкт проти 53,39 мс у Point Transformer і 344,62 мс у PointNet++. Загальні відмінності між моделями були невеликими, проте результати за окремими категоріями відрізнялися суттєвіше: Point

Transformer мав найвищий середній КЖ у восьми з шістнадцяти категорій, DGCNN – у п'яти, а PointNet++ – у трьох. Найбільший розкид значень КЖ між моделями спостерігався для категорій "мотоцикл" (Motorbike), "навушники" (Earphone) і "ракета" (Rocket) – саме на них вибір архітектури мав найбільше значення, тоді як категорію "ноутбук" (Laptop) усі моделі сегментували стабільно точно.

Висновки. Жодна архітектура не була найкращою за всіма критеріями. Point Transformer забезпечив найвищу якість сегментації, тоді як DGCNN показала найкраще виміряне співвідношення швидкості та якості. PointNet++ залишилася конкурентною за якістю, але мала значно більшу затримку в оціненій реалізації. Вибір моделі має враховувати поведінку на окремих категоріях, конфігурацію вхідних ознак та обчислювальні обмеження разом із загальними значеннями середнього КЖ.

Ключові слова: сегментація хмар точок, сегментація частин, ShapeNet Part, PointNet++, DGCNN, Point Transformer

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
02 September, 2026	20 September, 2026	22 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 57–76
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 004.4, 004.6, 004.7, 004.9

MODERN ARCHITECTURAL PRINCIPLES FOR DESIGNING HIGH-PERFORMANCE DISTRIBUTED SYSTEMS

Ivan Rovetskii
Lviv State University of Life Safety,
35 Kleparivska Street, Lviv, 79007, UKRAINE

Rovetskii, I. (2026). Modern Architectural Principles for Designing High-Performance Distributed Systems. *Electronics and Information Technologies*, 35, 57–76. https://doi.org/10.30970/eli.35.5

ABSTRACT

Background. The shift to online businesses has driven unprecedented growth in traffic density and data volumes. Engineering scalable high-load systems involves challenges like systemic latency, cascading failures from tight component coupling, I/O boundaries and CAP theorem trade-offs. Flawed architectural patterns cause performance degradation and financial losses. Therefore, the goal of this study is to analyze and systematize architectural solutions to minimize mentioned infrastructure challenges and ensure linear scalability of high-load applications.

Materials and Methods. This paper presents a theoretical study on the architectural design of high-load distributed systems based on Apache Kafka. The research synthesizes a comparative analysis of academic literature with the author's practical development experience.

Results and Discussion. A theoretical analysis reveals that synchronous inter-service communication based on HTTP/gRPC protocols leads to reduced efficiency in high-load environments due to cascading delays, tight coupling and thread pool exhaustion risks. The study substantiates that transitioning to an asynchronous Publish/Subscribe with a Pull-based model via Apache Kafka provides significant architectural advantages for extreme horizontal scalability. It provides a stable backpressure mechanism, prevents computational node overloading, minimizes CPU overhead through batch aggregation and simplifies broker architecture by shifting data offset persistence to the consumer. The paper systemically analyzes specific Apache Kafka engineering bottlenecks and proposes effective mitigation methods**.**

Conclusion. The paper analyzes architectural solutions for high-load systems based on the Apache Kafka platform, aimed at increasing the efficiency of stream data processing. The suggested approaches minimize the systemic risks of asynchronous communication, ensuring protection against event chronology violation in distributed transactions, data skew and anomalous message duplication.

Keywords: high-load distributed systems, CAP theorem**,** BASE model, Apache Kafka, backpressure, distributed transactions

INTRODUCTION

The rapid growth of data volumes and the necessity to ensure the uninterrupted operation of modern digital services require engineers to design high-load distributed software systems. The primary challenges in developing such systems are ensuring horizontal scalability, high throughput, and fault tolerance. Traditional monolithic architectures and synchronous communication (e.g., via REST/gRPC) demonstrate low efficiency under these conditions due to thread blocking and the formation of infrastructure bottlenecks. Consequently, asynchronous interaction based on the publish-subscribe

© 2026 Ivan Rovetskii. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

model has become the de facto standard for building modern systems. Apache Kafka [3] is the leading platform for implementing this model, which, owing to its pull-based consumption model, allows for the efficient distribution of workloads across computing nodes. However, the transition to asynchronous distributed architectures introduces complex engineering and academic challenges. Chief among these are the need for managing backpressure during peak workloads, as well as ensuring data consistency. Since classical two-phase transactions [4, 5] are inefficient in a distributed asynchronous environment, the research and adaptation of specialized architectural patterns become critically important.

Numerous fundamental works have been dedicated to the design of distributed systems and the optimization of message exchange. Specifically, the studies of M. Kleppmann [2, 5] thoroughly describe the trade-offs between data consistency and availability, while the works of the Apache Kafka platform developers, J. Kreps and N. Narkhede [3, 6], laid the foundation for distributed log processing. Architectural patterns for ensuring consistency in microservices, such as CQRS (Command Query Responsibility Segregation) and Event Sourcing, have been actively investigated in the works of G. Hohpe [1, 7] and E. Evans [8]. Variants of coordinating distributed transactions using the Saga pattern, along with mechanisms for guaranteed message delivery via the Transactional Outbox, are comprehensively detailed in the studies of C. Richardson [5, 9].

Despite the deep theoretical background and wide practical implementation of individual patterns, modern literature lacks a comprehensive analysis of their interaction under critical backpressure conditions. Most existing solutions examine the Saga or CQRS patterns in isolation, without accounting for the specific characteristics of the pull consumption model, where transaction processing delays on the consumer side can lead to an imbalance across the entire system. Consequently, there is a clear need for a systemic analysis and the formulation of recommendations regarding the effective combination of architectural patterns to minimize latency and preserve data integrity under high workloads. Therefore, the goal of this study is to analyze and systematize optimized architectural solutions based on Apache Kafka to ensure data consistency and backpressure resilience in high-load distributed systems.

MATERIALS AND METHODS

This paper presents a theoretical study on the architectural design of high-load distributed systems based on Apache Kafka, synthesizing a comparative analysis of scientific literature with the author's hands-on development experience.

RESULTS AND DISCUSSION

By the late 1990s, the ACID concept was the dominant paradigm in database engineering, requiring absolute transactional integrity and strict data consistency. However, with the rapid development of the early Web and the emergence of the first global search engines (primarily Google), developers realized that traditional relational databases were incapable of scaling horizontally across thousands of commodity servers. Attempts to maintain strict consistency across multiple servers over a network resulted in a catastrophic drop in performance and frequent system failures. In 2000, American computer scientist Eric Brewer presented a keynote at the Symposium on Principles of Distributed Computing (PODC), where he first formulated the CAP conjecture [10]. Brewer observed an empirical pattern: engineers building large-scale internet services were constantly balancing three parameters – Consistency, Availability, and Partition Tolerance – but no one succeeded in implementing all three simultaneously.

For two years, this statement remained merely a practical hypothesis until 2002, when scientists Seth Gilbert and Nancy Lynch from the Massachusetts Institute of Technology (MIT) published an official mathematical proof of Brewer's conjecture. Utilizing an asynchro-

nous network model where messages can be delayed or lost, they formally demonstrated that it is impossible to simultaneously guarantee both strict consistency and 100% availability in such an environment. From that moment, the conjecture achieved the status of the CAP theorem [11] – a fundamental law of distributed computing. It formalizes the mathematical and practical limits of distributed computing environments, asserting that any distributed system can simultaneously provide no more than two of the following three properties:

1. Consistency (C) – a guarantee that all cluster nodes see identical data at the exact same moment. Any read operation performed after a write operation must return the most recent value, regardless of which specific node the client queries.
2. Availability (A) – a guarantee that every request received by a functioning system node results in a non-error response, without the mandatory requirement that this response contains the latest version of the data.
3. Partition Tolerance (P) – the system's ability to continue stable operation despite network delays or communication losses between nodes.

In the context of real-world network infrastructure for high-load systems, the occurrence of communication anomalies, network latencies, and packet drops is a statistically inevitable regularity. According to the modern engineering paradigm, Partition Tolerance (P) is a constant that an architect cannot ignore. A distributed system lacking partition tolerance is a theoretical abstraction unsuitable for practical high-load operation. Consequently, the actual architectural choice narrows down to selecting between two trade-off configurations:

1. Consistency and Partition Tolerance (CP Model) – if a network segment becomes isolated, the system deliberately blocks read and write operations on unavailable nodes to prevent data desynchronization (Split-Brain). System availability is sacrificed to preserve absolute consistency (e.g., in financial transactions).
2. Availability and Partition Tolerance (AP Model) – in the event of a network partition, all nodes continue to accept requests and return responses. This maximizes system throughput and ensures an uninterrupted user experience; however, data across different cluster segments temporarily becomes inconsistent.

Engineers realized that in global networks, attempting to maintain 100% consistency makes the system slow and vulnerable to any network fluctuations. Efforts to preserve strict ACID (Atomicity, Consistency, Isolation, Durability) guarantees in high-load infrastructures resulted in a critical drop in performance due to the need for coordinating distributed locks (two-phase commits – 2PC). Under the constraints of the CAP theorem, attempting to maintain strict consistency (C) in a distributed network automatically destroys its availability (A). This technological contradiction drove an evolutionary transition toward an alternative architectural paradigm – the BASE (Basically Available, Soft State, Eventual Consistency) model proposed by Dan Pritchett [12]. BASE is the conceptual antipode to ACID: instead of pessimistic locking and instant consistency, it is built on the optimistic assumption that the system can exist in a state of temporary desynchronization to ensure continuous availability and linear horizontal scalability. Businesses learned to accept the fact that data across different servers might diverge for several milliseconds or seconds (e.g., user balances or warehouse inventory numbers updating asynchronously).

Unlike ACID systems, where a transaction is completely blocked and returns an error in the event of a local failure or network partition, an AP system within the BASE paradigm guarantees a mandatory response to the user's request. However, the quality of this response or the service itself can be subject to graceful degradation. The client receives a response that may contain partially stale data (e.g., displaying an old user balance or a cached version of a profile), or the system may temporarily disable non-essential functions (e.g., the personal recommendations block in e-commerce) while preserving the operability of the primary business scenario. The core architectural principle dictates that a successful, albeit imperfect, response takes precedence over a total system failure.

In classical DBMS, the system state is strictly deterministic and changes exclusively at the moment of a successful user transaction commit. The BASE model introduces the concept of a "soft" or dynamic state. This means that the data state across different cluster computing nodes can change independently over time, without direct interaction or operational input from the user or application code. The primary driver of these changes is the internal processes of asynchronous replication and log synchronization running in the background at the infrastructure level. During this synchronization window, different servers within the distributed system hold different versions of the same object, which is a normal technological state for high-load environments.

The Eventual Consistency property is the key stabilizing mechanism of the BASE model. It asserts that while the system abandons strict, instantaneous consistency, it guarantees that in the absence of new modification (write) operations on a specific object, data across all geographically distributed nodes and replicas will inevitably become identical and consistent after a certain, minimal possible time interval. This desynchronization time window (typically measured in milliseconds) allows the system to process colossal write streams without waiting for a simultaneous response from all servers in the cluster.

Thus, a systemic analysis of the evolution of distributed computing indicates that the transition from strict ACID guarantees to AP configurations within the BASE paradigm (specifically, the concept of Eventual Consistency) is a mathematically and engineering-justified step to achieve linear horizontal scalability. Abandoning instantaneous data consistency makes it possible to remove throughput limitations from storage subsystems and ensure continuous basic availability of digital platforms under extreme workloads. However, shifting data processing into a decentralized asynchronous environment and the necessity of independently maintaining eventual consistency at the application architecture level drastically complicates the software system lifecycle.

Systemic architectural anomalies and data consistency limitations in high-load distributed infrastructures

When millions of asynchronous operations execute concurrently without rigid coordination, a series of specific engineering anomalies emerges at the lower levels of infrastructure and network interaction. The practical implementation and operation of such systems are accompanied by the emergence of avalanche-like cascading failures of computing nodes, a critical accumulation of deadlocks during attempts to synchronize the soft state, and the throughput degradation of queues due to specific processing latencies of individual requests.

Let us examine in more detail the primary mechanisms underlying the issues that arise in high-load applications.

Cascading infrastructure failures.

In high-load microservice systems, components are interconnected via chains of synchronous requests. Each application server possesses a limited number of execution threads, managed within a thread pool. If one of the non-essential internal services (e.g., a report generation service or a third-party payment gateway) begins to delay responses due to database degradation or a network outage, the calling service receives no response, and its execution thread remains in a blocked state. Under intensive high-load traffic, all available threads of the calling service become blocked within seconds. Consequently, this service also stops responding to other components. This triggers an avalanche-like thread pool exhaustion, whereby a local failure of a single minor node cascade-fails the entire distributed system.

Concurrency-induced deadlocks.

When millions of users simultaneously interact with a system, the need to modify the same database records arises (e.g., stadium seat reservations, stock deductions, or wallet balance updates). To ensure transaction isolation, classical relational DBMSs utilize

pessimistic locking mechanisms (e.g., SELECT ... FOR UPDATE). If Transaction N1 locks Row A and attempts to access Row B, while concurrently Transaction N2 has already locked Row B and is waiting for Row A to be released, a state of mutual waiting occurs: a deadlock. In high-load systems, pessimistic locks generate massive wait queues at the DBMS level, leading to a sharp escalation in latency and the total failure of the storage subsystem.

Head-of-Line Blocking (HoL) anomalies.

This problem manifests in any environment governed by sequential data processing (FIFO – First In, First Out). If a "heavy" message (e.g., a request to generate a massive PDF report or process a media file array) enters a shared channel or queue, or if a message triggers an exception in the consumer code and continuously reverts to the front of the queue, the entire data stream stalls. Thousands of "lightweight" and fast messages (e.g., an order status update requiring only 2 milliseconds) queued behind this heavy request are forced to wait hours for its completion. This paralyzes the system's business logic and generates an artificial throughput deficit.

Under the constraints of the CAP theorem, given inevitable network latencies (Partition Tolerance), architects are compelled to adopt the BASE model. This marks a transition from strict instantaneous consistency (Strong Consistency) to eventual consistency. The issue of data consistency in high-load distributed architectures represents a fundamental challenge arising from the abandonment of monolithic databases and the transition to decentralized asynchronous microservices. Under extreme workloads, this leads to the emergence of stale or phantom data, concurrent write conflicts, and risks of losing transactional atomicity.

Desynchronization windows and Stale/Phantom data.

Since data replication across system nodes occurs asynchronously, a temporary consistency window (desynchronization time window) emerges. During this interval, different servers or read models hold conflicting versions of the same entity. For instance, when a user performs an action (e.g., purchasing an item), the record is successfully committed to the system's write model. However, upon an immediate page refresh, the corresponding read model has not yet processed the asynchronous event from the broker. Consequently, the client observes a stale system state (phantom data), which creates an illusion of a system failure or a lost transaction.

Concurrent write conflicts.

In a decentralized environment, there is no centralized arbiter (ACID coordinator) to enforce synchronous resource locking for the duration of a transaction. When millions of users interact with the system concurrently, scenarios involving the competitive update of a single entity inevitably occur. For example, two independent microservices might simultaneously read a user's balance (e.g., $100). Service A debits $20, while Service B concurrently debits $30. Due to the absence of synchronous row-level locking within the distributed network, both services will attempt to commit their respective local results ($80 and $70). As a consequence, one update will overwrite the other (the Lost Update problem), leading to a critical balance discrepancy and a complete failure of financial consistency.

Lack of distributed atomicity.

In distributed architectures, a single business process frequently spans multiple independent microservices, each managing its own isolated database (e.g., debiting funds in billing, reserving inventory in the warehouse, and accruing loyalty points). If the first two stages succeed, but the third service (loyalty points) encounters a crash or returns a validation error, the system is left in a fractured, inconsistent state. Funds remain debited, and inventory remains reserved, yet the overall business workflow is stalled. The classical ROLLBACK mechanism is entirely ineffective in such a network environment, as the first two local transactions have already been successfully committed within their respective databases.

Out-of-Order execution (Violation of causal ordering).

Asynchronous messages within distributed networks can experience latencies, undergo dynamic rerouting across different cluster servers, or be processed by parallel execution threads at varying speeds. For instance, if an event titled "Order Status Updated to Paid" is delayed in the network due to lag, while an event titled "Order Canceled by User" arrives faster, the consumer will process them in reverse chronological order. The system will then attempt to cancel an order that, according to its current local state, has not yet been created, and subsequently record a payment on an already terminated order – fundamentally destroying the domain's business logic.

Structural stabilization under Event-Driven Architecture (EDA).

All the mentioned destructive anomalies and data consistency challenges are successfully resolved within a modern Event-Driven Architecture (EDA) [17], where a distributed message broker serves as the primary actor in the distributed microservice system and functions as the central communication data bus. Transitioning from synchronous requests to an asynchronous stream of events allows for the complete decoupling of microservices. Instead of waiting for responses from adjacent nodes, systems commit business facts as immutable events that are instantly published to the broker. This entirely eliminates the risks of thread pool exhaustion and cascading infrastructure collapses.

To understand the mechanics of this operational shift, consider two contrasting examples of implementing the exact same business scenario: order processing on an e-commerce platform during peak traffic (e.g., Black Friday). The workflow spans four core components: Order Service -> Payment Service -> Inventory Service -> Notification Service.

When a customer triggers a purchase request, the Order Service allocates an execution thread and performs a synchronous HTTP call to the Payment Service, which in turn calls the Inventory Service. If a third-party banking gateway begins to experience latency at that microsecond, the Payment Service stalls, waiting on the network socket (Blocked state). Consequently, the execution thread of the Order Service remains blocked as well. Under extreme traffic spikes (e.g., 10,000 concurrent requests per second), the finite thread pool of the Order Service reaches total depletion within milliseconds (Thread Pool Exhaustion). The gateway slowdown cascade-fails the entire platform, rendering the core application completely unavailable to new users.

By leveraging an EDA coordinated by a distributed message broker, the Order Service executes a rapid, lightweight local transaction within its own database, instantly emits an OrderCreated event to the broker's topic, and immediately releases its execution thread to serve subsequent incoming requests. The thread retention time drops from several seconds to a mere 2–3 milliseconds. Operating autonomously, the Payment Service consumes the OrderCreated event from the broker. If the payment gateway degrades or crashes, it exerts zero impact on the Order Service, which continues to register new transactions unhindered. The events are safely buffered within the broker's persistent storage. Once the gateway's operation is restored, the Payment Service drains the accumulated queue at its own pace and emits an OrderPaid event, which is then asynchronously processed by the Inventory and Notification services.

Consequently, utilizing a message broker as a central data bus transforms temporary microservice outages or degradations from critical, system-wide failures into manageable, expected data-update latencies, ensuring continuous user interface availability for end users.

However, it is crucial to highlight that under extreme workloads, event-driven systems inherently face the destructive challenge of backpressure. This infrastructure-level anomaly manifests when the data production and invocation rates at the upstream producer level significantly outpace the nominal processing capacity and deserialization speed of the application logic on the downstream consumer side. In the absence of native regulation mechanisms, the unconstrained accumulation of unhandled event volumes triggers critical

volatile memory exhaustion (Heap Memory depletion), communication channel throughput degradation, and an avalanche-like collapse of the distributed infrastructure. The efficacy of mitigating this issue is directly dictated by the structural architectural model of interaction between the broker and its consumers, which classifies distributed environments into two fundamental paradigms: the push-based model and the pull-based model.

Architectural paradigms of data distribution. Analysis of push-based and pull-based consumption models

Under the push-based consumption paradigm, typified by traditional message brokers such as RabbitMQ [20], the broker cluster assumes absolute control over the data distribution vector. Upon receiving inbound payloads from producers, the broker autonomously, aggressively, and instantly pushes message packets onto the registered consumer instances via open TCP channels. This architectural behavior ensures minimal message delivery latency in low-to-moderate traffic conditions, as data is routed to available consumers without polling delays. However, under extreme workloads or sudden traffic spikes, this model strips the consumer nodes of operational autonomy. In a strict push scenario, the broker remains unaware of the downstream execution context or resource availability. Consequently, it continues to flood the consumer with data even if the application logic is stalled by heavy database transactions or thread blockages. Without explicit consumer-side intervention, this unconstrained transmission rapidly fills network buffers, saturates the consumer's local queues, and inevitably triggers catastrophic out-of-memory errors (Heap depletion) or node evictions. While RabbitMQ offers a mitigation framework via the Prefetch Count configuration (Quality of Service limits) – which allows consumers to declare a maximum threshold of unacknowledged messages to choke the push influx – this mechanism operates as a static infrastructure constraint rather than a dynamic, fluid stream regulation framework, shifting significant architectural complexity onto client-side resource budgeting.

However, RabbitMQ [20] is engineered as a transient message queue topology, where the physical replication engine is heavily reliant on volatile random-access memory (RAM). When the inbound traffic stream matches the processing velocity of the consumer nodes, messages reside exclusively in memory, ensuring ultra-low delivery latency. However, when backpressure triggers queue accumulation, RabbitMQ is forced to page unacknowledged payloads out of RAM and persist them onto physical disk storage via a destructive garbage-collected write pipeline. This paging mechanism induces severe computational overhead, triggers aggressive Erlang garbage collection loops, and drops the broker's throughput by up to an order of magnitude, resulting in volatile latency spikes and unpredictable cluster behavior. Furthermore, RabbitMQ utilizes index-heavy state tracking, where the broker cluster bears the full computational burden of maintaining message delivery acknowledgments and routing states for individual consumers. This design imposes a strict architectural limit on horizontal scaling, as expanding the cluster across multiple physical computing nodes requires intensive synchronization of queue states, creating a performance ceiling that typically tops out at 20,000 to 50,000 messages per second per queue under production workloads with strict persistence enabled.

Conversely, the pull-based data consumption model, fundamentally implemented at a high architectural level within the Apache Kafka [13-19]. distributed ecosystem, completely inverts the traffic flow control vector, offering a natural and resilient solution to the backpressure dilemma. Instead of allowing the broker to dictate data velocity, the pull paradigm grants complete operational autonomy to the downstream consumers. Under this framework, consumer microservices independently execute low-level data fetch routines (polling loops) to extract message batches from specific topic partitions. The data ingestion volume and transmission speed are regulated exclusively by the consumer node, strictly based on its available computing capacities, CPU allocations, and memory limits. If a consumer instance encounters downstream database degradation or performance

bottlenecks within its business logic execution, it simply delays or reduces the size of its subsequent poll requests. The broker cluster does not attempt to force the accumulated traffic; instead, it safely persists the inbound event stream within its immutable, sequential write-ahead logs on physical storage. This design shifts the state tracking burden away from the cluster coordinators and eliminates the risk of memory saturation or heap exhaustion across processing nodes. Consequently, the pull-based architecture transforms a potentially chaotic high-load surge into a controllable, deterministic data stream, neutralizing infrastructure-level stalls and providing a stable foundation for maintaining eventual consistency across highly scalable distributed platforms.

Let us formulate the primary architectural advantages of the pull-based streaming model implemented within the Apache Kafka ecosystem.

Log-centric storage paradigm

Unlike traditional message brokers that rely on destructive reads (deleting a message immediately after acknowledgement), Kafka operates as an immutable, append-only commit log. Messages are sequentially written to disk and persisted based on preconfigured retention policies (Data Retention). Since the consumption state is determined exclusively by a numerical pointer known as an offset, which is controlled directly by the consumer, it grants the consuming instance the capability to forcefully rewind its offset into the past and re-consume the historical event stream across any temporal window. This allows independent microservices to drain the same data stream at varying velocities without mutual lock contention. Concurrently, it provides the deterministic capability for event replay (Event Replay), which is foundational for Event Sourcing and CQRS patterns, and functions as a highly reliable transport layer for coordinating sequential steps within distributed transactions under high workloads. In this context, the Event Replayability mechanism in Kafka resolves two of the most complex challenges of Event Sourcing.

Deterministic state reconstruction and auditability after catastrophic failures

In traditional database management systems, recovering information after the corruption of the current system state is a complex, heuristic process. By utilizing Apache Kafka as an immutable, persistent event store (Source of Truth), any microservice can deterministically and flawlessly reconstruct its internal state from scratch. To achieve this, the consumer forcefully resets its offset to zero and sequentially processes the entire historical commit log. This additionally provides a 100% legally binding audit trail: the system is capable of mathematically replicating the exact state of the business domain at any specific microsecond in the past, which is critically important for highly regulated, high-load financial platforms.

Schema evolution and zero-downtime read model provisioning in CQRS

When a new business requirement emerges (e.g., deploying a new analytical microservice or altering the database schema of a read model), architects can entirely bypass complex data migration routines (ETL/Data Migration). The newly introduced service simply registers as an isolated, independent consumer group. It drains the Kafka topic from the absolute beginning (zero offset) in a background mode, independently constructing its own optimized read-side database (Read Model). This operational pipeline executes with zero impact on other running cluster components and achieves seamless system expansion without inducing operational downtime (Zero-Downtime).

The "Dumb Broker, Smart Consumer" paradigm.

Kafka completely eliminates computational overhead on the broker cluster side by shifting state management tasks to the consumer nodes. The broker bears no responsibility for tracking message delivery metadata for individual instances; instead, each consumer independently maintains its processing checkpoint within the log using a numerical pointer known as an offset. This allows for near-infinite scalability regarding concurrent worker distribution.

Native horizontal partitioning (Key-Based Partitioning).

Slicing logical topics into a multitude of independent physical segments (partitions) enables the linear horizontal distribution of I/O workloads across multiple cluster nodes. Utilizing cryptographic or non-cryptographic hashing of business keys, Kafka guarantees strict intra-partition FIFO order. The primary advantage for data consistency is that all events marked with the same business key (e.g., transaction ID) are guaranteed to be routed to the exact same physical partition. There, they are processed by a consumer strictly sequentially based on the FIFO (First-In, First-Out) principle.

This architectural capability elegantly isolates Head-of-Line (HoL) blocking anomalies and eliminates concurrent race conditions and concurrent write conflicts (the Lost Update problem) at the infrastructure level, without requiring the application of heavy distributed locks (two-phase commit – 2PC). Consequently, a message broker utilizing a pull-based polling model and partitioning transforms a chaotic asynchronous environment into a strictly ordered, fault-tolerant data stream.

Hardware efficiency and system resource optimization

Pull-based polling in Kafka achieves extreme throughput by minimizing context switches and CPU allocations. Leveraging the Linux kernel sendfile system call (Zero-Copy technology), the broker streams data bytes directly from the OS page cache into the network interface socket, completely bypassing user-space buffer allocation. This Zero-Copy payload transfer – routing data straight from the kernel cache to the network interface in bypass of user space – ensures a throughput capacity that outpaces push-based systems by an order of magnitude, closely approaching the physical limitations of the underlying hardware architecture.

Unlike push-based models that attempt to transmit each individual message independently – thereby inducing substantial CPU overhead due to excessive network packet header processing – a Kafka consumer can flexibly request data in large sequential batches (e.g., up to 500 records or with a 50 ms temporal window). This design pattern significantly minimizes network transaction frequency and optimizes system resource utilization.

In contrast to traditional asynchronous data buses, Kafka supports a built-in distributed transaction coordinator protocol to synchronize producer and consumer operations. This enables the atomic execution of the complete "read-process-write" loop across multiple distinct topics as an all-or-nothing operation, which is critical for preserving strict transactional consistency during the implementation of Saga and Transactional Outbox patterns under extreme workloads.

Saga pattern transforms a single large distributed transaction into a sequence of localized transactions executed within respective isolated microservices. Each saga step updates a local database and publishes an event to a message queue (Apache Kafka), which subsequently triggers the next execution step in an adjacent service. In the event of a business logic failure or a technical outage at any stage of the workflow, the Saga executes a reverse chain of so-called compensating transactions. These transactions sequentially roll back the modifications in reverse chronological order, returning the system to its initial consistent state [13, 19].

The transactional outbox pattern resolves the fundamental challenge of reliable message delivery by eliminating the dual-write problem between updating a service's local database and publishing an event to a message broker (Apache Kafka). Instead of executing two independent, non-atomic network requests, the business logic updates the primary domain tables and simultaneously writes the event to a specialized buffer table (outbox) within a single local ACID transaction in the service's database. A separate background process continuously polls new records from the outbox table and guarantees their delivery to the message broker with at-least-once delivery semantics. This completely eliminates the risk of event loss during sudden network partition events or application server restarts [13, 19].

Thus, Apache Kafka completely avoids the bottleneck associated with moving data between memory and disk (paging) by utilizing an append-only log model, which allows it to bypass user-space memory management mechanisms entirely. Instead of employing destructive read operations (where messages are deleted after acknowledgment), Kafka leverages the operating system's page cache via the Linux kernel's `sendfile` system call – a technique known as Zero-Copy. When a consumer requests a data packet, the broker transfers raw byte arrays directly from the OS page cache to the network interface socket, completely eliminating context switching and user-space memory buffer allocation. Through sequential data writing (Sequential I/O), Kafka avoids the costly random disk access operations that significantly degrade the performance of traditional database management systems; this enables physical throughput to scale linearly up to the limits of the Non-Volatile Memory Express drives and network interfaces being used.

In high-load enterprise systems utilizing modern hardware infrastructure (such as standard 10GbE or 25GbE networks combined with distributed multi-node clusters), a properly configured Apache Kafka system can deliver exceptionally high throughput, scaling seamlessly from 1 million to over 5 million messages per second with sub-millisecond latency. State tracking is elegantly offloaded to consumer clients via a numeric pointer (the offset)/ This means that adding thousands of parallel consumer processes incurs no performance degradation or additional memory overhead on the broker cluster, firmly establishing Kafka as the industry standard for high-throughput, horizontally scalable stream processing.

Engineering challenges of event chronology preservation and partition-level anomalies in Apache Kafka

Despite the high architectural efficiency and systemic advantages of the Apache Kafka distributed platform as a central data bus for CQRS and Event Sourcing models, its practical deployment in high-load environments is accompanied by a series of specific engineering challenges [13-19]. Ensuring extreme throughput and fault tolerance under the BASE paradigm requires architects to possess a profound understanding of the broker's internal mechanisms and to consciously manage design trade-offs. The first and most critical challenge within this context is guaranteeing a strict chronological order of event processing.

In event-driven architectures (EDA) and software systems implementing Saga or Event Sourcing patterns, the chronological sequence of messages is a critical constraint for business logic validity. Practical use cases requiring strictly guaranteed chronological ordering include*:*

- Financial operations and ledger balances (Banking). For example, debiting and crediting funds. If a credit notification event experiences network lag and arrives after a debit transaction event, the debit operation will trigger a "Failure: Insufficient Funds" exception, even though the necessary capital should have already been legally present in the account balance.
- Order management and inventory control (E-commerce)**.** For example, the sequential workflow of OrderCreated -> OrderPaid -> OrderCanceled. If the cancellation event bypasses the payment event due to network routing anomalies, the payment engine will process the transaction unaware of the termination, resulting in erroneous customer billing and permanent warehouse inventory locking.

Any disruption in event ordering triggers irreversible desynchronization of the distributed system's soft state, leading to critical application failures, logic loops, and financial liability risks. Within a distributed Apache Kafka infrastructure, enforcing strict message ordering is fundamentally complicated by its horizontally scalable, partitioned architecture and the intentional abandonment of synchronous distributed locking mechanisms.

Apache Kafka does not guarantee a global chronological ordering of messages across an entire topic. A global sequence exists exclusively within a specific physical partition commit log. If events relating to the lifecycle of a single entity (e.g., a specific order marked with order_id: 105) are written across different partitions of a topic, they will be processed concurrently by separate consumer instances, completely destroying the chronological event sequence.

To mitigate this constraint, the producing microservice is mandated to explicitly define a unique business entity identifier (such as an order_id or an account_id) as the message key during event publication. Kafka's native partitioning function (typically employing the MurmurHash3 algorithm) ensures that all events sharing an identical key are consistently routed to the exact same physical partition. This design constraint narrows the scope of the global ordering problem down to localized partition-level ordering, an environment where Kafka strictly enforces intra-partition FIFO sequence.

Even when correct key-based partitioning is applied, the chronological sequence of events can still be disrupted during data transportation from the producer to the broker. To maximize throughput, Kafka producers transmit messages asynchronously in sequential batches, leveraging internal memory buffers. If a producer dispatches Batch 1 (e.g., the OrderCreated event) and Batch 2 (e.g., OrderPaid) nearly simultaneously, and Batch 1 is dropped in transit due to transient network jitter, the broker returns an error response. Under a preconfigured automatic retry policy (retries > 0), the producer will retransmit Batch 1 However, during this network round-trip interval, Batch 2 may have already been successfully appended to Kafka's partition commit log. Consequently, within the physical partition, the payment event will be recorded ahead of the creation event, resulting in a critical out-of-order execution anomaly.

In high-load systems, a fundamental architectural contradiction emerges between the necessity of ensuring uninterrupted data processing and the requirement to maintain strict event ordering within a distributed commit log. Given that Apache Kafka guarantees chronological sequence (FIFO) exclusively within a single physical partition, a severe engineering challenge arises during exception handling and fault mitigation.

This problem possesses a dual nature and manifests as two interconnected operational anomalies:

1. Infrastructure-level partition stalling
2. Chronological inversion induced by the deployment of asynchronous non-blocking retry mechanisms.

If, during sequential data extraction from a partition, a specific record (e.g., Event 2) triggers an exception within the business logic due to transient downtime of adjacent databases, network timeouts, or payload defects (a "poison pill"), the consuming microservice cannot commit its processing offset forward. The instance is compelled to cyclically re-execute processing attempts for the exact same message (the blocking retries mechanism). Because the execution thread is bound by Event 2, the consumer is incapable of advancing to read subsequent messages (Events 3, 4, etc.) queued behind it within the same partition log. Consequently, a localized processing failure of a single business entity completely paralyzes asynchronous replication and read-side model replication for thousands of other concurrent users whose events happened to be routed to this specific physical partition via key-based hashing.

To circumvent this Head-of-Line blocking effect and ensure the continuous velocity of the primary queue, engineers frequently implement a non-blocking retries pattern. Under this topology, if a message (Event 2) encounters a processing failure, the consumer forcefully extracts it from the primary stream, reroutes it to an isolated retry topic or a Dead Letter Queue (DLQ), subsequently commits the primary topic offset forward, and instantly proceeds to the next record. Once the anomalous Event 2 is shifted into the retry pipeline, the consumer immediately drains Event 3 from the primary log and successfully completes

its execution. After a preconfigured backoff timeout, a separate retry handler activates and eventually processes Event 2 successfully.

However, as a direct consequence of this infrastructure maneuver, strict chronological execution order is completely violated: the business logic for Event 3 is applied to the system state earlier than that for Event 2. Such a chronological inversion is fundamentally unacceptable for maintaining data consistency within CQRS and Event Sourcing architectures.

Architectural mitigation mechanisms for event chronology preservation and stream integrity within Apache Kafka

The examined dilemma between infrastructure-level partition stalling (Head-of-Line Blocking) and chronological sequence degradation induced by non-blocking retry topologies (Retry Topics / DLQ) mandates that architects deploy highly specialized defensive design patterns. Attempting to resolve these operational anomalies strictly at the message broker's baseline configuration level is fundamentally ineffective. Consequently, safeguarding data model integrity under the constraints of eventual consistency shifts from the infrastructure layer directly to the application logic and component-level architecture. Within this framework, several advanced architectural mechanisms can be delineated to effectively neutralize the mentioned vulnerabilities, thereby ensuring end-to-end stream protection from the upstream publisher down to the final read-side data store.

Producer-level idempotence configuration (Idempotent Producer Protocol).

The first line of defense for preserving event chronology involves mitigating operational anomalies that manifest during data transit from the source service to the message broker due to network fluctuations and automated retry policies. At the producer client library level, the strict idempotence protocol is activated via the enable.idempotence=true configuration. Upon initialization, the Kafka coordinator cluster assigns a globally unique Producer ID (PID) to the producing instance, alongside a monotonically increasing Sequence Number to each discrete message batch targeted at a specific partition.

If a network jitter event prevents the producing instance from receiving an execution acknowledgment for a successfully committed Batch 1, the retry mechanism triggers retransmission. The broker intercepts this call and evaluates the incoming Sequence Numbers against its state ledger. If Batch 1 is already persisted within the log, Kafka automatically discards the duplicate execution request. Furthermore, the broker strictly enforces contiguous sequence validation: if a retry anomaly causes Batch 2 to arrive ahead of the delayed Batch 1, the broker rejects the out-of-order write attempt. This guarantees that events are appended to the partition's physical commit log in a strict chronological FIFO sequence, fully maintaining downstream network throughput without inducing message reordering risks.

Key-based ordered retries and cascading backoff topologies.

To leverage the throughput benefits of non-blocking retries – namely, freeing the primary partition log from execution stalls caused by anomalous payloads – while preventing chronological inversion for causally related events, a specialized architecture of multiple ordered retry topics is deployed. Instead of routing all failing payloads into a single monolithic retry queue, the design implements a cascading pipeline of sequential retry topics (e.g., topic-retry-1, topic-retry-2, ..., topic-dlq), each governed by a progressive exponential backoff timeout.

When Event 2, bound to a specific entity (entity_id: 105), encounters an unhandled exception due to transient downstream dependency failure, it is rerouted to topic-retry-1. Simultaneously, the consumer locks this specific business key within a high-performance distributed cache (e.g., Redis), marking it as "temporarily blocked." When the subsequent Event 3, belonging to the exact same business entity (entity_id: 105), is immediately drained from the primary topic log, the consumer checks the key status in Redis. Detecting

an active ordering conflict, the consumer bypasses local execution logic and sequentially routes Event 3 into the retry pipeline immediately behind Event 2. Concurrently, if a subsequent Event 4 linked to an independent entity (entity_id: 200) arrives, it is processed successfully with zero operational delay. Consequently, intra-entity FIFO chronology is strictly maintained on a per-instance basis, while the overall data velocity of the primary partition remain completely unhindered.

Parallel consumption with entity-key level locking (Parallel Consumer Pattern).

A more elegant and highly performant mitigation method deployed directly at the consumer client layer involves utilizing multi-threaded processing mechanics within the boundaries of a single physical partition, a concept programmatically realized via the Kafka Parallel Consumer architecture. Shifting away from the classical pipeline where a solitary consumer thread sequentially processes events from an assigned partition, the microservice drains a massive event batch from Apache Kafka and delegates it to an internal thread pool execution engine.

Prior to task allocation, the consumer's internal orchestrator aggregates the extracted messages based on their business entity identifiers (partition keys). Events linked to distinct keys are dispatched to separate CPU worker threads and processed concurrently, thereby optimizing hardware resource utilization. Conversely, events sharing an identical business key (e.g., the historical lifecycle sequence of a single user account) are strictly bound to the same execution context and processed sequentially. If Event 2 for a specific key stalls due to localized retry cycles, the internal orchestrator isolates and locks the execution loop for that specific entity key alone. Events belonging to all other independent keys within the same batch continue to be processed concurrently across parallel threads.

Offsetting and state tracking updates back to the Kafka broker cluster are managed via a sophisticated moving sliding-window acknowledgment scheme (per-message acknowledgment tracking). This infrastructure capability ensures maximum stream velocity and throughput while fully eliminating data loss risks during partial failure scenarios.

Engineering of idempotence at the read-side model layer

Given that any architectural maneuvers involving message rerouting, automated retries, and network partition failures inevitably lead to duplicate data delivery under at-least-once delivery semantics, the final line of systemic defense involves guaranteeing consumer instance idempotence. Read-side databases within the CQRS architectural pattern (e.g., Elasticsearch, Redis, or Cassandra) are engineered such that the redundant application of the exact same event does not alter the system's final state ledger.

Depending on the maturity and structure of the inbound event stream, the process of dynamic message uniqueness verification by the consumer node is executed via two distinct operational pipelines:

1. Explicit unique identifiers – if the upstream producer injects a unique system-level identifier (such as an event_id formatted as a UUID) or a unique transaction token (transaction_id) into each message payload, the computational overhead of generating a hash from the entire message body is avoided. The consumer directly utilizes this pre-existing string token as an explicit uniqueness key.
2. Deterministic payload hashing – in the absence of end-to-end unique identifiers, the consuming microservice independently extracts a cluster of critical business fields whose combination defines the operational uniqueness of the transaction over a specific temporal window (for instance, in a financial transaction: sender_id + recipient_id + amount + timestamp_minutes). These targeted fields are serialized into a strict, sequential byte array, from which a deterministic hash is generated utilizing fast, non-destructive algorithms (e.g., SHA-256 where cryptographic collision resistance is mandatory).

Within high-load infrastructures, significant architectural value is achieved through a hybrid approach combining explicit entity identifiers with payload hashing. If an incoming

message possesses a stable entity_id (e.g., a user instance identifier) but lacks a clear state version number, the consumer computes a hash strictly from the remaining payload fields representing the object's current state attributes. The resulting composite key (entity_id + payload_hash) enables the system to verify whether an automated retry cycle is attempting to re-apply an absolutely identical modification state to the same business entity during network spikes. This mitigates redundant write cycles and eliminates unnecessary I/O locking bottlenecks within the CQRS read-side storage engine.

Regardless of the mechanism utilized to extract the unique tracking key (whether via a direct UUID string or a dynamically generated deterministic payload hash), prior to executing any transactional write operations within the CQRS read-side database, the consumer microservice performs an atomic checkpoint evaluation. This is achieved by attempting to write the extracted key into a high-performance distributed caching cluster (typically Redis) utilizing the atomic SETNX (Set if Not Exists) command protocol, accompanied by a strict time-to-live parameter (TTL, e.g., 24 hours).

If the Redis coordinator returns a success status, it verifies that the incoming message is being processed by the system context for the first time. The projector service then unhindered executes the corresponding CQRS read model modification query, commits the consumer offset forward to the Apache Kafka cluster, and terminates the processing iteration loop. Conversely, if the Redis node rejects the write operation – indicating that a key containing an identical identifier or payload hash has already been registered within the state ledger – the consumer instantly identifies the current event as an redundant delivery induced by automated retry policies or network jitter spikes. The microservice completely bypasses the execution of heavy business logic computations and storage engine write cycles; instead, it marks the event payload as successfully processed, advances the Kafka consumer offset pointer forward, and seamlessly transitions to the next transaction.

Leveraging explicit unique identifiers maximizes deduplication velocity while completely eliminating the mathematical vectors of hash collisions. On the other hand, generating a deterministic payload hash in the absence of explicit identifiers enables the implementation of robust idempotence layers even across heterogeneous, unstructured data streams. Utilizing volatile In-Memory caching via Redis guarantees that key validation latency is restricted to sub-millisecond thresholds, which is well within acceptable limits for high-throughput, high-load environments.

However, the primary operational challenge within this topology remains the configuration and control of the TTL parameter (the deduplication temporal window). Architects must carefully balance the volatile memory capacity constraints of the Redis cluster infrastructure against strict business domain requirements regarding the maximum retention lifespan for delayed stale events arriving from downstream retry pipelines.

The Data Skew anomaly and corresponding architectural mitigation strategies

The distributed architecture of Apache Kafka is fundamentally built upon the principle of parallelism, where extreme throughput is achieved by evenly distributing the message traffic across a multitude of independent physical partitions. However, within real-world high-load infrastructures, maintaining absolute traffic symmetry encounters a fundamental engineering anomaly known as data skew (or partition skew).

As established previously, to enforce strict intra-partition FIFO chronological ordering, producing instances leverage consistent partitioning algorithms based on business key hashing. However, the probability distribution of these transaction keys in a real-world enterprise ecosystem is almost never uniform. A classic manifestation of this issue occurs in large-scale B2B platforms or macro-marketplaces, where a single enterprise-tier client account or top-tier merchant generates millions of transactions per hour, whereas thousands of micro-users emit only isolated, sporadic requests. Because Kafka's native partitioning hash function rigidly binds a specific business key to a dedicated partition, an operational crisis unfolds: a single physical partition of a topic absorbs up to 80–90% of the

entire high-load ingestion volume, while the remaining cluster partitions remain heavily underutilized. This phenomenon is technically defined as data skew [13-19].

Data skew triggers a cascade of destructive infrastructural consequences, primary among which is the effective neutralization of horizontal scalability benefits. Even if a topic is scaled across 64 physical partitions and the corresponding consumer group scales out to 64 independent server instances, 63 of those nodes will remain idle or underloaded. Concurrently, the solitary consumer thread allocated to the saturated partition will operate at the absolute thermal and physical thresholds of its CPU capacity and volatile memory limits, rendering the expanded infrastructure ineffective.

This architectural imbalance directly causes a critical escalation of consumer lag within the processing pipeline. Because a single execution thread is physically incapable of draining the gigabyte-scale impulsive influx hitting the overloaded partition log, event processing latency escalates from milliseconds to minutes or hours. Under the constraints of eventual consistency, this latency creates long-lasting data desynchronization windows within the CQRS read-side models for that entire user segment.

Furthermore, this disruption results in asymmetric storage depletion across broker nodes within the cluster layout. The specific cluster brokers hosting the leader replicas of the saturated partitions begin to rapidly accumulate massive write-ahead logs. This highly uneven disk utilization can prematurely trigger aggressive cluster data retention policies, causing the forced truncation of logs and the catastrophic loss of unconsumed message payloads.

To overcome the destructive impact of data skew during the design phase of high-load infrastructures, software architects deploy specialized trade-off strategies aimed at redistributing traffic across the cluster.

One primary approach involves synthetic key expansion, commonly known as the Key Salting Technique. If a specific business key is identified as anomalously large, such as vendor_id_555, the producer dynamically appends a randomized or round-robin synthetic suffix – the "salt", ranging from vendor_id_555_1 to vendor_id_555_4 – prior to data transmission. This architectural modification forces Kafka's native partitioning function to scatter the transaction volume of a single massive client evenly across multiple distinct physical partitions. However, this strategy introduces a significant architectural trade-off, as absolute chronological ordering across the entire entity scope is sacrificed and preserved only within individual salted sub-partitions, which consequently shifts the burden of ordering and deduplication onto the downstream CQRS read-side models.

Another effective pattern is the implementation of a two-stage distributed aggregation topology, which is primarily utilized within stateful streaming analytics frameworks such as Kafka Streams or Apache Flink. Under this design, the inbound stream data is initially partitioned via a completely randomized key mapping, intentionally bypassing strict ordering rules to perform primary, low-level in-memory metric pre-aggregation. This initial phase drastically decreases the physical traffic volume before subsequent routing. Following this pre-aggregation step, the analytical results are remapped and forwarded into the target topic utilizing the final, deterministic business key, successfully preserving cluster stability without overwhelming individual nodes.

Finally, architects utilize composite key routing to balance the data distribution across the computing environment. This method involves engineering complex, multi-attribute partition keys that encapsulate not only the unique entity identifier but also a granular temporal marker or a transaction typology attribute, such as vendor_id + transaction_type. Decomposing the key space in this manner allows the system to partition different event types of a single enterprise-tier client across separate physical computing nodes, thereby preventing single-partition saturation and maintaining smooth horizontal scalability.

The Consumer Group Rebalancing anomaly and corresponding architectural mitigation strategies

The high fault tolerance and operational elasticity of Apache Kafka are natively sustained by its consumer group architecture. The broker cluster dynamically coordinates the allocation of physical topic partitions among available instances of consuming microservices. However, under high-load conditions, this partition redistribution workflow – technically defined as rebalancing – mutates into a severe engineering bottleneck capable of inducing massive throughput degradation and critical processing delays (*Consumer Lag*). The cluster transaction coordinator (a designated Kafka broker node) initiates a rebalancing routine whenever changes occur within the group topology or topic metadata configurations:

- Dynamic horizontal scaling – the hot plug-in deployment of new consumer nodes to handle peak influx volumes.
- Planned or unplanned node demise – scheduled rolling restarts or abrupt infrastructure crashes of a service instance.
- Topic topology alterations – modifying the total partition count within a live topic metadata ledger.
- False node eviction – the most prevalent failure vector under extreme workloads, where a consumer instance remains physically viable and healthy but fails to process a massive accumulated data batch within the preconfigured temporal threshold (max.poll.interval.ms). The cluster coordinator presumes the service is stalled, forcefully evicts it from the active topology ledger, and declares a cluster-wide rebalance.

Under legacy infrastructural configurations, such as utilizing the traditional Eager Rebalance Protocol, this state synchronization process triggers a cascade of critical operational anomalies [13-19].

The first major disruption manifests as the "Stop-the-World" synchronization stall, during which an eager rebalance routine forces the cluster coordinator to revoke partition allocation privileges from all consumer instances within the active group concurrently. As a consequence, inbound data ingestion and processing freeze entirely across the global pipeline for several seconds or even minutes while node synchronization and partition reassignment states are resolved. Under extreme high-load conditions, even a sub-second processing pause triggers the accumulation of hundreds of thousands of unhandled message payloads within the broker's persistent buffers.

This synchronization freeze directly leads to exponential processing lag escalation, creating a severe lag spike upon rebalance finalization when the consumer group suddenly encounters a massive, aggregated volume of accumulated message traffic. Forcing the system to aggressively drain this backlogged data generates extreme CPU resource saturation across consuming instances and induces volatile load impulses within downstream read-side engines, such as the Query DB in CQRS architectures.

Furthermore, this disruption frequently causes transactional duplication and state reconciliation storms across the cluster topology. If a consumer instance successfully executes a processed data batch but the group transitions into a rebalance state before the execution offset commitment is synchronized back to the Kafka broker registry, a critical inconsistency unfolds. The newly provisioned node assigned to that specific partition will resume data ingestion from the last successfully recorded historical checkpoint, triggering massive redundant message execution loops. This duplication heavily saturates application-level idempotence layers and severely drags down the overall performance of the asynchronous CQRS read-side data synchronization pipeline.

To minimize the destructive impact of rebalancing operations within high-load environments, enterprise software architects deploy targeted mitigation configurations and design patterns across the system architecture. Primary among these methods is the deployment of the cooperative sticky assignor protocol. Shifting away from the legacy

Eager Rebalance Protocol, which mandates a global and indiscriminate revocation of all partition allocations, the Cooperative Sticky Assignor mechanism leverages an incremental state synchronization algorithm. Under this framework, the cluster transaction coordinator targets and migrates exclusively those specific physical partitions that require relocation to alternate nodes, allowing the remaining consumer instances to continue data ingestion and processing unhindered from their designated logs, thereby completely neutralizing the catastrophic "Stop-the-World" synchronization stall.

In parallel, the implementation of node identity persistence via the Static Membership Pattern serves as a critical mechanism to prevent unnecessary group state disruptions. Consumer instances are explicitly allocated a globally unique, stable deployment identifier using the group.instance.id configuration parameter, which is typically mapped to a deterministic Kubernetes Pod name string. When a microservice node restarts, such as during a rolling code deployment cycle, and successfully recovers within the preconfigured temporal window defined by session.timeout.ms, the cluster coordinator recognizes the node as a pre-existing participant. Consequently, a group-wide rebalance is bypassed entirely, and the physical partition mapping remains strictly anchored to that specific computing instance.

Finally, clusters must be immunized against false node evictions by ensuring the strict decoupling of system polling loops from heavy computational latencies. This defensive pattern involves definitive thread isolation and lifecycle decoupling, where the primary consumer execution thread exclusively runs low-level data fetch routines via the poll method. This thread instantly offloads the drained message batches into an internal, asynchronous Worker Thread Pool dedicated to business logic execution, immediately returning to its next sequential iteration loop. This strict segregation guarantees that the client instance consistently transmits cluster health heartbeats back to the broker coordinator registry within mandated boundaries, fully independent of the execution complexity or temporal duration of individual Saga transactions.

CONCLUSION

As a result of the conducted study and the comprehensive systematization of architectural principles for constructing high-load scalable software architectures, several primary outcomes have been achieved.

First, the efficacy boundaries of legacy architectures were substantiated. Research has shown that monolithic topologies and synchronous interaction protocols (such as HTTP/REST and RPC) exhibit limited effectiveness under extreme loads, as the rapid exhaustion of execution resources and thread pools, combined with network I/O latency, significantly hinders the maintenance of system resilience.

Second, the study systematically demonstrates that traditional push-based message streaming models exhibit severely limited horizontal scalability under extreme workloads. A rigorous architectural evaluation of push-based frameworks, exemplified by RabbitMQ, reveals that aggressive, broker-driven data distribution causes an uncontrollable influx of message packets during sudden traffic spikes, effectively stripping consumer nodes of operational autonomy. Furthermore, the analysis highlights that push-based architectures suffer from significant computational overhead induced by heavy index-based state tracking and message delivery acknowledgment management on the broker side, alongside the performance degradation triggered by memory-to-disk paging and disruptive garbage collection loops during backpressure events, which collectively establish a strict infrastructure throughput ceiling.

Third, a rigorous comparative analysis demonstrated that deploying Apache Kafka represents a sound and engineering-appropriate architectural choice for high-load infrastructures. Its advantages over legacy message brokers and traditional queuing networks are validated by inherent backpressure regulation, maximized concurrency, and

universal persistence semantics. Owing to the pull-based ingestion model, downstream consumer nodes independently regulate traffic ingestion velocity, effectively mitigating volatile memory saturation and heap exhaustion risks. Furthermore, horizontal topic partitioning combined with low-level sequential disk operations and Zero-Copy payload transfer technology enables linear hardware resource utilization up to the physical boundaries of underlying network card interfaces and storage arrays. Additionally, long-term message commit log retention coupled with client-side offset tracking provides a unique event replayability capability, which is critically important for the background re-population of CQRS read models and deterministic state reconstruction within Event Sourcing topologies.

Finally, the operation of Apache Kafka under extreme workloads reveals specific operational anomalies that require targeted counteraction. In the context of addressing intra-partition FIFO violations and Head-of-Line (HoL) blocking, the paper proposes the integration of the Idempotent Producer Protocol, explicit key-based message routing, and multi-threaded Parallel Consumer architectures as potential structural solutions. Within the context of alleviating partition data skew and traffic asymmetry, the study proposes synthetic key space expansion via the Key Salting Technique alongside two-stage distributed streaming aggregation topologies as viable mitigation options. Furthermore, to overcome the challenges of consumer group rebalancing, the paper proposes transitioning to the incremental Cooperative Sticky Assignor Protocol, implementing static node membership registries, and enforcing strict thread-level decoupling of system polling loops from application logic. Lastly, in the context of managing redundant message delivery during retry cycles, the study proposes high-performance uniqueness filtering using explicit UUID event tokens and dynamic payload-based hashing combined with atomic In-Memory distributed state caches as possible read-side remedies.

The formulated recommendations and architectural mitigation frameworks demonstrate that a highly reliable configuration of the Apache Kafka ecosystem serves as a key infrastructural determinant for the successful engineering of fault-tolerant, high-performance digital platforms across diverse data-intensive domains, including E-commerce networks, FinTech architectures, IoT ecosystems, and Big Data processing infrastructures.

Future advancements within this research domain will focus on conducting empirical benchmarking experiments and executing precise latency metrics measurement to perform a comprehensive comparative analysis between the Key-Based Ordered Retries architecture and the Parallel Consumer pattern deployed across cloud-native environments managed under Kubernetes cluster orchestration.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author received no financial support for the research, writing, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The author declares that he has no competing interests.

The author has read and agreed with the published version of the article.

REFERENCES

[1] Newman, S. (2021). *Building microservices: Designing fine-grained systems* (2nd ed.). O'Reilly Media.

[2] Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media.

[3] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the 6th International Workshop on Networking Meets Databases (NetDB 2011)* (pp. 1–7). https://people.csail.mit.edu/matei/courses/2015/6.S897/readings/kafka.pdf

[4] Özsu, M. T., & Valduriez, P. (2020). *Principles of Distributed Database Systems (*4th ed. pp. 183–246*).* Springer. https://doi.org/10.1007/978/3-030-26253-2

[5] Richardson, C. (2018). *Microservices Patterns: With examples in Java*. Manning Publications.

[6] Narkhede, N., Shapira, G., & Palino, T. (2017). *Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale*. O'Reilly Media.

[7] Hohpe, G., & Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley.

[8] Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.

[9] Garcia-Molina, H., & Salem, K. (1987). Sagas. In *Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data* (pp. 249–259). ACM. https://doi.org/10.1145/38713.38742

[10] Brewer, E. A. (2000). Towards robust distributed systems (abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing* (p. 7). ACM. https://doi.org/10.1145/343477.343502

[11] Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News, 33*(2), 51–59. https://doi.org/10.1145/564585.564601

[12] Pritchett, D. (2008). BASE: An acid alternative. *ACM Queue, 6*(3), 48–55. https://doi.org/10.1145/1394127.1394128

[13] Henning, S., & Hasselbring, W. (2024). Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud. *The Journal of Systems and Software*, 208, 111879. https://doi.org/10.1016/j.jss.2023.111879

[14] Wang, G., Chen, L., Dikshit, A., Gustafson, J., Chen, B., Sax, M. J., Roesler, J., Blee-Goldman, S., Cadonna, B., Mehta, A., Madan, V., & Rao, J. (2021). Consistency and completeness: Rethinking distributed stream processing in Apache Kafka. In *Proceedings of the 2021 International Conference on Management of Data* (pp. 2602–2613). ACM. https://doi.org/10.1145/3448016.3457556

[15] Menda, J. R. (2018). Real-time financial settlement using Kafka Streams and Cassandra: A distributed architecture for low latency, exactly-once processing. *Journal of Scientific and Engineering Research*, *5*(10), 362-372. https://doi.org/10.5281/zenodo.18084995

[16] Bonér, J. (2016). *Reactive microservices architecture: Design principles for distributed systems*. O'Reilly Media. https://www.oreilly.com/library/view/reactive-microservices-architecture/9781491975664/

[17] Vernon, V. (2023). *Strategic Monolith-to-Microservices Migration Using Domain-Driven Design and Event Streaming*. Addison-Wesley.

[18] Vogel, A., Henning, S., Pérez-Wohlfeil, E., Ertl, O., & Rabiser, R. (2024). A comprehensive benchmarking analysis of fault recovery in stream processing frameworks. *In Proceedings of the 18th ACM International Conference on Distributed and Event-Based Systems* (pp. 171–182). https://doi.org/10.1145/3629104.3666040

[19] Stopford, B. (2024). *Designing Event-Driven Systems: Concepts and Patterns for Streaming Services with Apache Kafka*. O'Reilly Media.
[20] Videla, A., & Williams, J. (2012). *RabbitMQ in action: Distributed messaging for everyone*. Manning Publications.

СУЧАСНІ АРХІТЕКТУРНІ ПРИНЦИПИ ПРОЕКТУВАННЯ ВИСОКОНАВАНТАЖЕНИХ РОЗПОДІЛЕНИХ СИСТЕМ

Іван Ровецький
Львівський державний університет безпеки життєдіяльності,
вул. Клепарівська 35, Львів, 79007, УКРАЇНА

АНОТАЦІЯ

Вступ. Стрімкий розвиток цифрової економіки спричинив зростання обсягів даних та щільності трафіку. Проектування високонавантажених платформ часто супроводжується системними затримками, каскадними збоями інфраструктури через жорстку зв'язність компонентів, а також обмеженнями пропускної здатності підсистем вводу-виводу. Помилки при виборі архітектурних шаблонів призводять до деградації продуктивності та фінансових втрат. Метою даного дослідження є аналіз та систематизація сучасних архітектурних принципів для мінімізації зазначених інфраструктурних викликів та забезпечення масштабованості високонавантажених застосунків.

Матеріали та методи. У роботі виконано теоретичне дослідження особливостей проектування архітектури високонавантажених розподілених систем на основі брокера повідомлень Apache Kafka, яке базується на аналізі та порівнянні фактів, отриманих з наукових джерел, а також у результаті отриманого практичного досвіду автора.

Результати. У результаті теоретичного аналізу виявлено, що синхронна міжсервісна взаємодія на основі протоколів HTTP/GRPC призводить до зниження ефективності у високонавантажених середовищах через каскадні затримки, жорстку зв'язність та ризик вичерпання пулів потоків. Обгрунтовано, що асинхронна модель за шаблоном «Публікація/Підписка» та модель запитів даних споживачем на основі брокера Apache Kafka надає архітектурні переваги для забезпечення горизонтального масштабування, запобігаючи перевантаженню вузлів. У роботі також проаналізовано інженерні виклики, які виникають під час експлуатації брокера Apache Kafka.

Висновки. У роботі проаналізовано архітектурні рішення для високонавантажених систем на основі брокера Apache Kafka, спрямовані на підвищення ефективності процесів потокового опрацювання даних. Запропоновано підходи, які мінімізують системні ризики асинхронної комунікації, забезпечуючи захист від порушення хронології подій у розподілених транзакціях, нерівномірного розподілу трафіку та аномального дублювання повідомлень.

Ключові слова: високонавантажені розподілені системи, CAP-теорема**,** BASE-модель**,** Apache Kafka, зворотний тиск, розподілені транзакції

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
06 September, 2026	16 September, 2026	21 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 77–96
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.75, 004.41

A REVIEW OF CLOUD-BASED AUDIO PROCESSING USING DIRECTED ACYCLIC SIGNAL-PROCESSING GRAPHS IN MICROSERVICE ARCHITECTURES

Oleh Chaplia* & *Halyna Klym*
Lviv Polytechnic National University,
12 Stepan Bandera Street, Lviv, 79013, Ukraine

Chaplia, O., & Klym, H. (2026). A Review of Cloud-Based Audio Processing Using Directed Acyclic Signal-Processing Graphs in Microservice Architectures. *Electronics and Information Technologies*, 35, 77–96. https://doi.org/10.30970/eli.35.6

ABSTRACT

Background. Audio processing chains in digital audio workstations (DAWs) are directed acyclic graphs (DAGs) of digital signal processing (DSP) nodes. Executing such graphs in the cloud promises elastic compute, collaboration, and access from thin clients. Microservices dominate cloud systems, but their principles were shaped by request-and-response workloads, not by block-synchronous audio. This paper reviews approaches that process and render audio in the cloud as DSP DAGs and evaluates them against microservice principles.

Materials and Methods. A systematic protocol was applied to two complete proceedings volumes of the IEEE International Symposium on the Internet of Sounds, using database searches and backward snowballing. From 97 proceedings papers and 49 full texts from other sources, 43 peer-reviewed studies published from 2022 to 2026 were included and grouped into eight method families, and each method was assessed on execution location, service granularity, real-time mode, reported latency and its measurement point, scaling, fault tolerance, and support for DAW-originated graphs.

Results and Discussion. Browser DAWs keep the DAG on the client and use the cloud for storage and collaboration. Networked music performance systems relay streams within strict latency budgets but do not render offline graphs remotely. Edge rendering services move a single stage to the network edge. Audio microservices expose offline tasks through REST APIs. Cloud-native media pipelines run containerized graphs with self-healing, and generic DAG engines provide scheduling and scaling but ignore sample-accurate timing. None of the reviewed systems execute a DAW-originated DSP DAG as microservices with block-level deadlines. Reported latencies are measured at different points and are not comparable. Cloud and container systems report hundreds of milliseconds to seconds.

Conclusion. Existing systems lack a portable representation of DAW graphs, a granularity model for stateful DSP nodes, deadline-aware scheduling in container orchestrators, low-latency block transport between services, and benchmarks for cloud audio rendering. These gaps define the research agenda for microservice-based cloud audio processing.

Keywords: cloud computing, microservices, digital signal processing, directed acyclic graph, digital audio workstation, reliability

INTRODUCTION

Audio production, broadcasting, game audio, virtual reality, and music education rely on chains of digital signal processing (DSP) operations, and in a digital audio workstation

© 2026 Oleh Chaplia & Halyna Klym. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

(DAW) such a chain is a graph. Tracks feed insert effects, sends feed auxiliary buses, buses feed the master chain, and the master chain feeds a render or a playback device. Because feedback loops are not allowed inside the processing chain, the graph is a directed acyclic graph (DAG). The DAW engine sorts the graph topologically and executes it block by block. The same computation model is used by web audio engines, modular synthesis environments and media frameworks [1, 2]. Immersive mixes for object-based and binaural delivery add object renderers and bed buses to the same graph, and practitioners already manage such sessions in their DAWs [3].

Cloud computing has changed how such systems are built and delivered. The Internet of Sounds vision describes networks of sound devices that sense, process, and exchange audio with the help of edge and cloud resources [1]. Remote and cloud-based production has moved from proposals to deployed systems, as the review of remote immersive audio production and the cloud-based delivery systems in the broadcast literature documents [4, 5]. Remote immersive audio production extends this trend to many channels transported to a remote mixing site and to shared virtual spaces with many concurrent listeners [4]. Browser-based DAWs have reached a level where multitrack recording, plugins, and offline rendering work without a native installation [2]. At the same time, the dominant architectural style for cloud software is the microservice architecture. Microservices offer independent deployment, horizontal scaling, technology heterogeneity, and fault isolation [6, 7]. These qualities are attractive for a cloud audio engine that must serve many users and many sessions at the same time.

However, audio DSP is a poor fit for the assumptions behind classic microservices. A block of 128 samples at 48 kHz lasts 2.67 ms. Musicians who play together over a network tolerate a mouth-to-ear delay of 20 to 30 ms [8]. A cloud microservice call, a container restart, or a serverless cold start takes tens to thousands of milliseconds. This is the tension that any cloud audio engine must resolve. DSP nodes such as filters, delays, and reverbs hold internal state that must follow the stream. Every remote edge in the graph adds a network hop. Retries, circuit breakers, and autoscaling reactions operate at time scales that are much longer than a block period. A cloud audio system therefore needs a careful mapping between the DSP DAG and the set of services that execute it.

Several research directions touch this problem from different sides. GPU-oriented DAW APIs execute audio graphs on accelerators [9]. Web Audio Modules bring a plugin standard to browser DAWs [2]. Microservice backends were used to add generative music functions to a DAW through a VST plugin [10], and DSP as a service was proposed as a general model for server-based music production [11]. Edge computing was applied to spatial audio rendering [12]. Cloud native media pipelines were containerized and orchestrated in Kubernetes [13]. Generic DAG engines for serverless and stream processing workloads mature quickly [14, 15]. Two recent works cover neighboring ground. Turchet et al. survey the Internet of Sounds as a whole, including its edge and cloud layers [1]. Turchet and Krstulovic propose the vision of DSP as a service, which moves music production tasks to servers, segments such services into asynchronous, reactive, and low-latency tiers, and analyzes three prototypes under latency and reliability [11]. Neither takes the microservice architecture or the DAW origin of the graph as its object. Still, no review brings these works together and asks how a DSP DAG that originates from a DAW can be executed as a set of cloud microservices.

Remote and cloud production workflows are now documented in peer-reviewed work rather than only in vendor material [4, 5]. Browser DAWs reached multitrack recording, plugin hosting, and offline rendering between 2023 and 2026 [16, 2, 17]. Container orchestration and serverless platforms matured to the point where containerized media graphs run in production with self-healing [13]. The building blocks exist, but the mapping between them and a DSP DAG has not been studied.

This paper fills this gap with three contributions.

1. A taxonomy of eight solution families for cloud, edge, and browser-based execution of audio DSP graphs, derived from 43 peer-reviewed studies selected with an explicit protocol.
2. A mapping of eight microservice principles to the requirements of DSP DAG execution, which classifies each principle as conflicting, neutral, or requiring an audio-specific interpretation.
3. A research agenda of eight gaps, ranked by the number of included studies that touch each gap.

The goal is to review existing solutions for processing and rendering audio in the cloud as DSP DAG graphs, to describe their models, methodologies, advantages and drawbacks, to compare them, and to identify what such systems lack today. The focus is on the microservice architecture and its principles applied to audio DSP graphs, including graphs that originate from DAW sessions. The rest of the paper is organized as follows. The Materials and Methods section describes the review method. The Results and Discussion section presents the solution families, the comparison, the mapping of microservice principles to DSP DAG requirements, the answers to the research questions, the open problems, and the threats to validity. The Conclusion summarizes what is missing and outlines further research.

MATERIALS AND METHODS

Review protocol

This study is a systematic literature review with a narrative synthesis. The protocol defines the sources, the search strings, the inclusion and exclusion criteria, the quality checklist, and the data extraction fields before the search. Three search stages were used. Stage 1 screened two complete proceedings volumes, the 2023 4th International Symposium on the Internet of Sounds and the 2024 IEEE 5th International Symposium on the Internet of Sounds, because this venue concentrates the peer-reviewed work on networked, edge, and web-based audio. Stage 2 searched the publisher sites of IEEE Xplore, ScienceDirect, Springer Link, ACM Digital Library, MDPI, the Journal of the Audio Engineering Society, and the SMPTE Motion Imaging Journal using a general web search engine, and checked Scopus indexing for every candidate on the publisher page. The search strings combined one term from the first group with one term from the second group. The first group was "cloud", "microservice", "serverless", "edge computing", "container", "Kubernetes" and "DAG". The second group was "audio", "DSP", "audio graph", "digital audio workstation", "networked music performance", "spatial audio rendering", "media pipeline" and "stream processing". Stage 3 applied backward snowballing to the reference lists of the included full texts. The search was performed in September 2026.

The inclusion and exclusion criteria are the following.

1. Included. Journal articles indexed in Scopus, or papers in IEEE or ACM conference proceedings, published from 2023 to 2026.
2. Included. The paper describes, implements, or evaluates a system, architecture, protocol, or model for processing, rendering, or transporting audio, or for executing processing graphs, in a cloud, edge, container, browser, or networked setting.
3. Excluded. Preprints, vendor documentation, white papers, magazine articles, and product announcements.
4. Excluded. Conference papers not published by IEEE or ACM.
5. Excluded. Papers on music information retrieval, perception, or datasets without a system architecture component.
6. Excluded. Papers whose full text was not accessible to the authors, so that the extraction fields could not be verified.

One exception to the time window was accepted. The journal article by Bianchi et al. from 2022 was included through snowballing because it is the only work that formalizes the DAW graph for parallel execution [9]. Each included paper was checked against a quality

checklist with four items. The venue is peer-reviewed. The system or model is described in enough detail to classify it. An evaluation or a measurement is reported, or the paper is a position or review paper that is marked as such. The reported latency, if any, has a stated measurement point. Papers that failed the second item were excluded. The other items were recorded and are used in the comparison.

For every included paper, the following fields were extracted into a table. Solution family, DAG location, whether the DAG is explicit, service granularity, real-time mode, reported latency with its measurement point, scaling mechanism, fault tolerance mechanism, DAW origin, and the research gaps the paper touches. The screening and the extraction were performed by one reviewer. The family assignment was done by open coding of the extracted fields followed by grouping into families, and the grouping was revised twice until every paper fitted one family. Then an extraction table with all fields and the quality checklist outcome per study, and the study-to-gap matrix that produces the gap ranking were created.

Selection results

Fig. 1 shows the selection flow. Stage 1 screened 97 proceedings papers by title and abstract, 47 from the 2023 volume and 50 from the 2024 volume. 36 papers met the topic criteria. 19 of them were excluded because their full text was not accessible, so 17 were included, and 80 were excluded in total. 61 were excluded for music information retrieval, perception studies, and datasets without a system component, and 19 for missing full-text access. Stage 2 ran 31 query executions that returned about 300 result entries, which were screened by title and snippet, and 49 records were assessed in full text. 21 records were included, and 28 were excluded: 9 as preprints, 5 outside the time window, 3 as workshop or magazine articles, 3 as vendor material, and 8 for missing full-text access. Stage 3 added 5 records by backward snowballing after 2 candidates were excluded for missing full-text access. In total, 43 peer-reviewed studies were included. 17 come from the two proceedings volumes and 26 from journals and other IEEE or ACM proceedings. All 43 passed the first two checklist items. 31 report an evaluation or a measurement, and 12 are

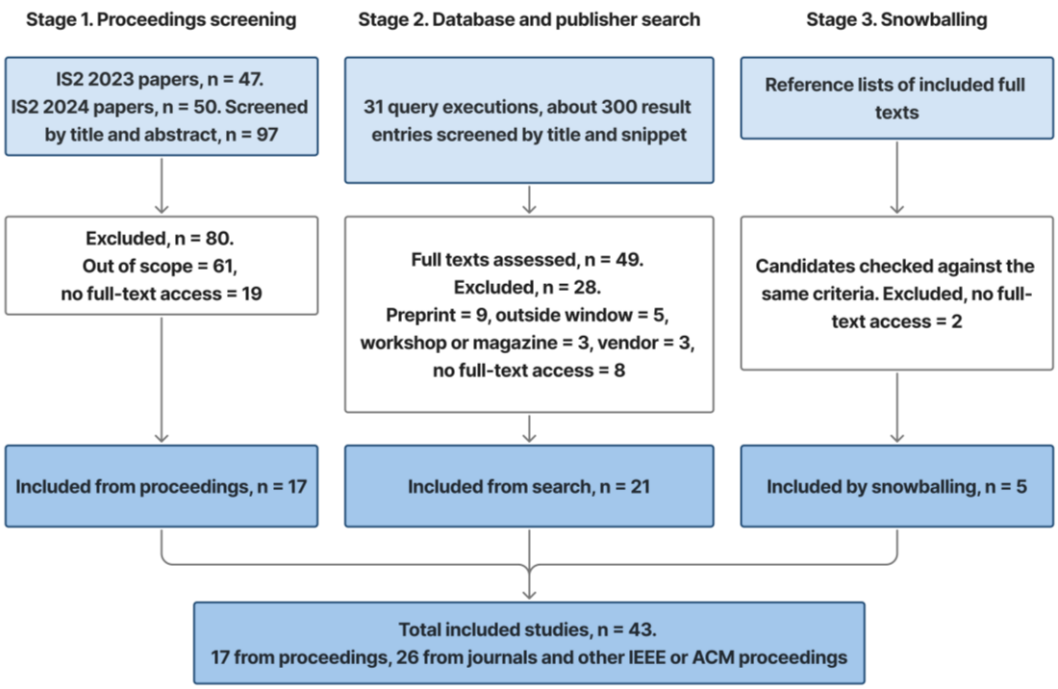


Fig. 1. Selection flow of the review with the number of records at each stage

position, review, or design papers. 24 reports a latency together with its measurement point. **Fig. 2** shows the distribution of the included studies over the eight solution families and the three sources. The hatched segments show that the audio microservice, media pipeline, and generic DAG engine families come entirely from outside the symposium, whereas the fog family comes entirely from it.

Research questions

The review answers the following research questions.

1. RQ1. Which existing solutions process or render audio in the cloud, at the edge, or in the browser as a DSP DAG, and which computation model do they use?
2. RQ2. Which of the microservice principles conflict with block-synchronous execution of a DSP DAG, for which principles does the literature contain a workaround, and which principles apply without change?
3. RQ3. Which latencies do the solution families report, at which point were they measured, and are they comparable across families?
4. RQ4. What is missing for executing a DAW-originated DSP DAG as cloud microservices, and how many included studies touch each missing element?

Comparison criteria

The comparison criteria are the following.

- DAG location. Client, edge, cloud, or a combination.
- Explicit DAG. Whether the solution represents the processing chain as an explicit graph that can be inspected, serialized, and scheduled.
- Service granularity. Monolith, one service per task, one service per DSP node type, or one service per subgraph.
- Real-time mode. Block-synchronous processing, streaming with a latency budget, or offline rendering.
- Reported latency, its measurement point, and the measurement method.
- Scaling and fault tolerance mechanisms.
- Support for DAGs that originate from a DAW session or a DAW plugin format.

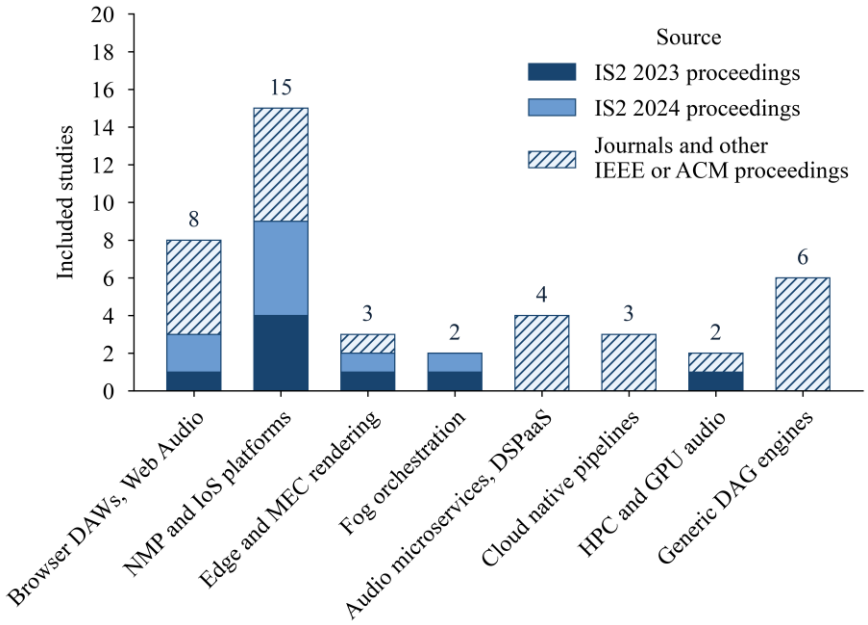


Fig. 2. Distribution of the 43 included studies over the solution families and the three sources of the review

RESULTS AND DISCUSSION

The eight solution families are presented from the client to the cluster. Browser DAWs execute the graph on the client. Networked music performance systems keep the graph on the client and move only streams. Edge and fog systems move one stage to the network edge. Audio microservices, media pipelines and HPC systems execute in the cloud or on a cluster. Generic DAG engines close the section because they execute graphs in the cloud without an audio deadline. Before the families, the computation model itself is described.

DSP DAG graphs as a computation model and their origin in DAWs

A DSP DAG is a dataflow graph with synchronous semantics. Each node consumes and produces a fixed number of samples per block, so the schedule of one block is static and can be computed once when the graph changes. This property is used by DAW engines, by the Web Audio API rendering graph, by GStreamer pipelines, and by domain-specific languages such as Faust that compile block diagrams to efficient code [2, 13]. The GPU-oriented API for DAWs proposed by Bianchi et al. formalizes the DAW project as a graph of tracks and effects and maps its independent branches to parallel GPU kernels, which confirms that the DAW graph is a suitable unit for parallel and distributed execution [9]. Real-time DAG scheduling theory studies the same structure for multicore embedded systems, including response time bounds along the critical path and the choice between global and partitioned scheduling [18]. Two quantities decide whether a distributed graph meets its budget: the block period, which is the block size divided by the sample rate, and the cost of the critical path, which is the sum of node processing times and network transfers along the longest path from a source to the output.

Fig. 3 shows a small DAW session as a DSP DAG. Audio and MIDI clips are sources. Inserts such as compressors, equalizers, gates, and sound synthesizers are intermediate nodes. Sends route a copy of a signal to an auxiliary reverb bus. The sum bus and the master limiter are the last stages before rendering. The graph is acyclic, but it is not a tree, because sends create additional paths. The longest path from a source to the output defines the latency floor of the session. **Fig. 4** shows how such a graph maps to a microservice deployment. A gateway authenticates the client and enforces quotas. A graph compiler validates the graph, partitions it into subgraphs, and places them on node-type services. A block transport layer connects the services. Supporting services keep node state, store stems and renders, and collect observability data.

The DAW origin of the DAG matters for two reasons. First, the DAW already stores the graph, the plugin identifiers, the parameter automation, and the routing in its project file. A cloud audio engine can therefore obtain the DAG from the DAW instead of asking the user to rebuild it. Second, DAW plugins follow established formats such as VST, AU, CLAP, and, for the web, Web Audio Modules [2, 10]. A cloud node service that hosts such a plugin inherits its processing semantics, its parameter model, and its state model. This

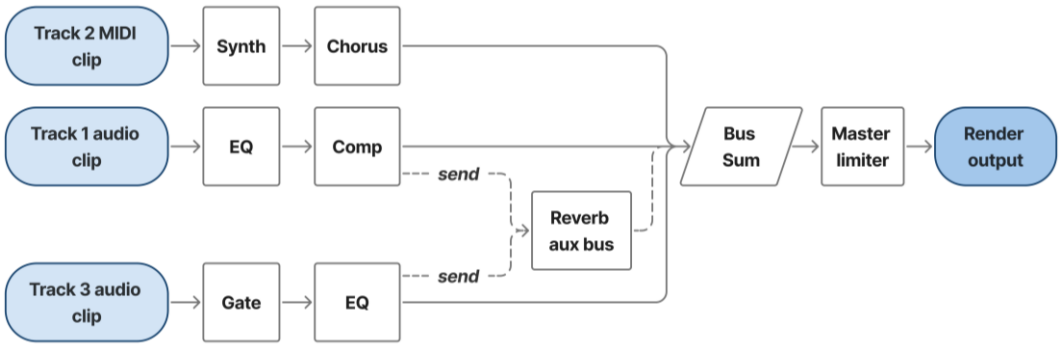


Fig. 3. A DAW session represented as a DSP DAG

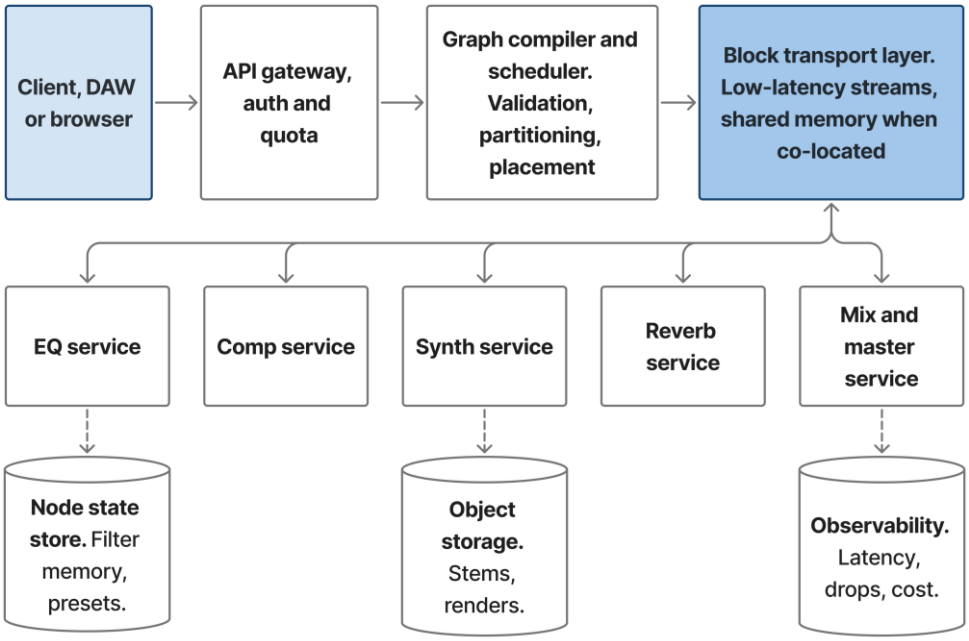


Fig. 4. A DAW session mapped to microservices

makes the plugin host the natural unit of a node service. It also brings licensing, sandboxing, and binary compatibility problems into the container, because most plugin formats were designed for a single desktop process.

Representation formats for audio objects and scenes exist at several levels, but none of them describes a processing graph. Turchet and Antoniazzi describe a semantic web of musical things that achieves interoperability between devices through ontologies and service descriptions [19]. Laitinen et al. present the OMASA format of the 3GPP IVAS codec, which jointly codes audio objects and a metadata-assisted spatial audio scene for immersive mobile communication at 13.2 to 512 kbps [20]. These formats describe devices, objects, and scenes, not DSP nodes with parameters and per-block state, which is what a cloud engine needs. Also, deep learning nodes are entering audio graphs. ANIRA is an architecture for neural network inference inside real-time audio applications that moves inference to a separate thread and respects the block deadline [21]. Such nodes are heavy and stateful, which makes them the first candidates for placement on remote services.

Browser-based DAWs and Web Audio graphs

Browser-based DAWs are the closest existing systems to a cloud audio engine from the user point of view. WAM-Studio is an open-source DAW that uses Web Audio, Web MIDI, WebAssembly, and Web Components together with the Web Audio Modules plugin standard. Buffa and Vidal-Mazuy introduce it as a multitrack workstation for the web with plugin hosting and round-trip latency handling [16], and Buffa and Demont report on latency management, synchronized audio and MIDI recording, MIDI tracks, and offline rendering of a project [2]. The DAG in this system is the Web Audio rendering graph. Its nodes are native browser nodes and WebAssembly worklets. The graph executes on the client. The cloud stores projects, samples, and plugins and enables collaboration.

Other web-based systems use the same approach with a server that coordinates many browser clients. Matuszewski and Golvet describe rapid prototyping of distributed musical things with web technologies, where a server coordinates many browser clients that each run their own audio graph [22]. Buffa, Girard and Hofr use Web Audio Modules for immersive audio collaboration in the musical metaverse, where plugins run in every

client, and a server synchronizes the shared state [23]. Turchet and De Cet describe a web-based distributed system that integrates mobile devices of the audience in a choral performance [24]. Boem, Tomasetti and Turchet compare two-dimensional and mixed-reality interfaces for remote music making and find that usability, not immersion, decides adoption [25]. Gil Panal, David and Richard present Hi-Audio, an open-source web DAW for asynchronous multitrack recording and dataset distribution with a Flask REST backend, server-side annotation with machine learning models, server-side compression, and a browser round-trip latency estimation based on maximum-length sequences, whose variability reaches a standard deviation of about 8 ms in Chromium browsers [17].

The advantages of this family are zero installation, a standard graph model, a plugin standard, and mature low-latency execution on the client. The drawbacks are client device limits, the lack of server-side DSP execution, and the absence of a service model. Hi-Audio is the only system of the family with server-side processing, and it applies it to annotation and compression after upload, not to the audio graph [17]. Collaboration is implemented at the project level and not at the audio block level. The DAG is explicit, but it is not portable to a server-side engine because Web Audio nodes exist only in the browser. From the microservice point of view, these systems are monolithic clients with a thin cloud backend.

Networked music performance and Internet of Sounds platforms

Networked music performance (NMP) systems connect musicians over a network. Turchet and Rottondi relate NMP to ubiquitous music and to the Internet of Musical Things and summarize the latency constraint of 20 to 30 ms mouth-to-ear delay [8]. Turchet and Tomasetti identify the latency contributions of immersive NMP systems, including capture, encoding, transmission, spatial rendering and playback, and relate them to available hardware and software [26]. Turchet and Casari assess a private 5G standalone and a public 5G non-standalone architecture for NMP and show which network configurations satisfy the latency budget [27]. The Internet of Sounds review by Turchet et al. places these systems in a broader agenda that includes cloud and edge processing, semantic interoperability, and sustainability [1]. Turchet and Casari also evaluate satellite links for NMP in rural areas, which extends the reach of such systems at the cost of higher and more variable delay [28]. Tomasetti and Turchet show that musicians prefer binaural audio with head tracking over stereo when playing with others through headphones, which adds a spatial rendering stage to the audio path [29].

NMP systems are relevant here because they set the latency budget for any cloud audio system that is used in a live setting. Their architecture, however, is a relay. A server or a peer forwards streams and, at most, mixes them. The DSP graph of each musician stays on the local machine. Packet loss concealment, jitter buffers, and clock synchronization are the main engineering topics. Vignati, Dallona and Turchet provide a modular testbench to study packet loss concealment algorithms [30]. Three works move the transport into hardware or dedicated radio to remove software jitter: an FPGA-based audio coprocessor [31], Ethernet audio transmission directly to an FPGA [32], and DECT-2020 NR radio for professional live audio [33]. The common finding is that the transport can be made deterministic when it is owned end to end, which a cloud service mesh does not offer. Real-time embedded platforms such as Elk Audio OS show that deep learning models can be executed inside the audio callback with deterministic latency, but again on a single device [34].

The advantage of this family is a well-studied latency model with perceptual thresholds and measurement methods. The drawback is that the DSP DAG is not executed remotely, so the results cannot be transferred directly to a cloud engine. Microservice principles appear only in the signaling and session management parts of these systems.

Edge and MEC-based audio rendering

Edge computing brings compute close to the user and reduces the network delay of remote edges. Martusciello et al. implement a spatial audio service on a multi-access edge

computing (MEC) 5G infrastructure and analyze its performance [12]. The rendering stage runs on the edge server, and the client receives the rendered stream. The 3GPP immersive voice and audio services codec targets the same edge, using the OMASA format to carry separate talker objects along with a parametric scene captured by a mobile device, so rendering can happen at the endpoint or in the network at a low bitrate [20].

Fog-based orchestration extends this idea to many devices. Mikolajczyk et al. present gestural interaction with object-based audio in an Internet of Sounds ecosystem, where audio objects and their rendering are distributed across the ecosystem [35]. Vieira et al. formulate design guidelines for immersive environments that combine musical, multimedia, and multisensory things [36].

Remote production is the closest use case to a cloud audio engine in this family. Transporting many immersive channels to a remote mixing site is the main cost, which Giacomelli et al. document in their review of remote immersive audio production, along with the improvements they propose [4]. None of these works exposes the renderer as a graph of services.

The advantage of this family is the explicit trade-off between placement and latency, and the use of standard telecommunication infrastructure. The drawback is the granularity. The rendering stage is one service. The DAG inside the renderer is fixed and not exposed. Placement is static and chosen by the operator. Scaling to many concurrent sessions and fault tolerance are not evaluated, and only two of the three works in this family report a latency measurement. Still, these works provide the first measurements of rendering a DSP stage remotely with a network in the loop, which is exactly when network transfer enters the critical path.

Microservice systems for music and audio tasks

A small number of works apply microservice architecture directly to music and audio functions. Raposo and Soares design a modular cloud architecture that adds generative music capabilities to a DAW through a VST plugin [10]. The backend consists of a gateway microservice with API key validation, a database microservice, a generative harmony microservice implemented in Java, and a chord-to-MIDI microservice implemented in Python, all exposed through Node.js REST services. The JUCE-based plugin runs inside the DAW, requests chord progressions, and drags the generated MIDI into the timeline. The measured computation time is about 865 ms for generation and about 1300 ms for MIDI conversion, and the network transfer time is negligible in comparison. The authors state that the latency is acceptable for composition but not for live performance, and that concurrency and containerized scaling remain future work.

Turchet and Krstulovic generalize this direction into a vision of DSP as a service (DSPaaS), in which musical devices connected to the Internet delegate production tasks to servers [11]. They identify the building blocks, the technological enablers and the network configurations of such a system, and they segment its applications into three tiers by latency constraint. Asynchronous services answer in seconds or more, reactive services within a few hundred milliseconds, and low-latency services within 10 to 30 ms or below, which is the threshold at which a musician no longer perceives the delay between an action and its sound. Three prototypes illustrate the tiers, among them a MIDI pattern recognition service in which a Raspberry Pi with Elk Audio OS sends recorded patterns over TCP/IP to a remote GPU server. The prototypes are analyzed under the key performance indicators of latency and reliability. The vision names the design trade-offs of a cloud audio service more precisely than any other reviewed work, but the server is treated as one service. Neither a processing graph nor a decomposition into services appears in it.

Work on microservices provides the architectural background for such systems. We proposed a framework for automated code scaling and execution on multiple distributed machines, where a primary service shares the user code with replica services and executes it across the network without an orchestrator [7]. An improved organization of REST API handlers that reduces project complexity and improves the maintainability of Node.js

microservices was proposed [6]. These results apply to the control plane of a cloud audio engine, that is, to gateways, session management, graph storage, and health monitoring.

The advantage of this family is a clean application of microservice principles, including gateways, independent services, and language heterogeneity. The drawback is that the services operate on requests and files, not on audio blocks. The DSP chain is absent because the services produce MIDI, code, or metadata, or it is hidden inside one server, and none of the systems executes a DAG with block-level deadlines. The three latency tiers of DSPaaS are the most useful classification of cloud audio services in the reviewed set, and only the third tier is a live audio graph problem [11]. The VST integration of Raposo and Soares is the only case where the DAW is the client of cloud microservices, and it uses the DAW as a user interface rather than as the source of the graph [10].

Cloud native media processing pipelines

Media processing frameworks represent pipelines as graphs of elements with negotiated formats, which is the same model as a DSP DAG. Herranz Sanchez et al. present Voctomix 2.0, a cloud native architecture for real-time remote video production based on a GStreamer pipeline [13]. The mixing core, the ingest processes, and the operator interface are containerized. The architecture was evaluated on a native host, in a Docker Compose stack of eleven services, and in a single-node Kubernetes cluster. Containerization and orchestration added no appreciable processing overhead. The median command-to-output latency was 293 ms at 25 frames per second, which is determined by the frame period. Forced failures of a camera ingest were recovered in about 1.8 s by the restart policy, while a stream blanker kept the program output valid.

This work is the most complete evaluation of a containerized real-time media graph in the reviewed set. Three lessons transfer to audio. First, uncompressed internal traffic should stay inside a shared network namespace, because raw media between containers over a physical network is expensive. Second, self-healing by the orchestrator works at the scale of seconds, so output continuity needs a concealment mechanism in the graph itself. Third, telemetry should be decoupled from the processing path with asynchronous messaging. The drawback for audio is the time scale. A video frame period of 40 ms is an order of magnitude longer than an audio block period, and the audio path in this system is a single stereo stream that follows the video.

Related media systems confirm the pattern. Del Rio et al. deliver quality-of-experience-aware live media with a containerized processing chain for user-generated content [37]. Fukudome and Nishimura describe a cloud-based linear video delivery system that uses edge computing on a content delivery network [5]. Both systems operate at the frame or segment level. Neither reports an audio path with a block-level deadline.

High performance and GPU acceleration for audio graphs

Turchet, Vella and Fiore analyze the potential of high-performance computing for the Internet of Sounds and propose a compute continuum where edge devices perform light feature extraction, network nodes aggregate, and HPC clusters with GPUs and FPGAs perform heavy rendering, such as spatial audio for thousands of concurrent users [38]. They identify the communication overhead between CPU and GPU and the deadline of the audio callback as the main obstacles for real-time use. Bianchi et al. show that a GPU-oriented API can execute the DAW graph in parallel on the GPU with acceptable latency, provided that the graph is analyzed and batched correctly [9].

The advantage of this family is throughput. A single accelerated host can render many graphs or very large graphs. The drawback is that the execution stays on one host and does not use a service model. Scheduling on the GPU is a task graph problem, and the results are directly useful for a node service that hosts an accelerated subgraph. The idea of a compute continuum also suggests a natural partitioning rule. Nodes with strict latency stay near the client, and nodes with heavy but tolerant computation move to the cluster.

Generic DAG execution in cloud and serverless systems

Outside the audio domain, the execution of DAGs in the cloud is a mature research topic. Henning and Hasselbring benchmark the scalability of five stream processing frameworks deployed as microservices on Kubernetes, with more than 740 hours of experiments and up to 110 concurrent service instances [14]. All frameworks scale approximately linearly when resources are provisioned, but they differ in the rate at which resources must be added. Stream processing operators are stateful, run continuously, and communicate through message logs, which is closer to audio processing than request-based services. Their latency, however, is measured in tens of milliseconds to seconds and is dominated by the message broker. Fragkoulis et al. trace the evolution of stream processing systems and show that state management, exactly-once semantics, and elasticity are the dominant design goals, not bounded per-record latency [39].

Serverless platforms execute functions on demand and compose them into workflows. Raith, Nastic and Dustdar review serverless edge computing and identify placement, cold starts and stateful functions as open problems [15]. A cold start is incompatible with a block deadline of a few milliseconds unless the function is kept warm for the whole session. Real-time DAG scheduling theory, reviewed by Verucchi, Olmedo and Bertogna, provides response time analysis and schedulability tests for DAG tasks on multicore platforms under global and partitioned scheduling [18]. These results are formulated for a single machine with shared memory, so the network transfer delay between nodes is not part of the model. On the orchestration side, Dakic, Kovac and Slovinac evaluate Kubernetes for high-performance workloads with machine-learning-based placement [40]. Placement quality improves, but the scheduler does not see deadlines.

This family offers a rich set of scheduling, autoscaling, and benchmarking methods. The drawback is the absence of sample-accurate deadlines and of an audio transport. The frameworks optimize throughput and cost, whereas an audio graph needs bounded latency for every block. Still, the partitioning heuristics, the placement models, and the benchmark methodology of these works can be adapted to DSP DAGs.

Comparison of the reviewed solutions

Table 1 compares the eight solution families against the criteria defined in the Materials and Methods section. The families cover different points of the design space. Browser DAWs have an explicit DAG and a DAW origin, but no server-side execution. NMP and edge systems have real-time execution with a network in the loop, but no explicit DAG and no service decomposition. Audio microservice systems have service decomposition, but no DAG and no real-time mode. Cloud native media pipelines and generic DAG engines have both an explicit graph and an orchestrated deployment, but they lack the DAW origin and the audio block deadline. No reviewed system combines all four properties, namely an explicit DSP DAG, a DAW origin, block-level real-time execution, and a microservice deployment.

The measurement point column shows why the reported latency numbers are not comparable across families. Browser and NMP works report buffer or one-way network delays. Edge works report motion-to-sound delays. Microservice works report request processing times. The media pipeline reports command-to-output delay. There is no common benchmark, no common workload, and no common measurement point. This is a gap in itself, because a cloud audio engine must be evaluated along a defined critical path, with a defined block size and a defined network topology.

The reported figures of the cloud and container families, 293 ms command-to-output and 1.8 s recovery for the media mixer [13] and 0.8 to 1.3 s per request for the DAW microservice backend [10], lie two orders of magnitude above the 20 to 30 ms live budget of networked music performance [8], whereas the browser, NMP and edge families report values inside that budget but for a single device or a single stage. Because the cloud and container figures describe a request or a command and not a single audio block, this gap

Table 1. **Comparison of solution families for cloud, edge and browser-based processing of audio DSP graphs**

Solution family	DAG location	Explicit DAG	Service granularity	Real-time mode	Reported latency	Measurement point	Scaling and fault tolerance	DAW origin
Browser DAWs and Web Audio graphs [16, 2, 23, 17]	Client	Yes, Web Audio graph	Monolithic client, cloud for storage	Block-synchronous on the client	Milliseconds	Audio buffer on the client	Limited by the client device	Native, web DAW
NMP and IoS platforms [8, 26, 27, 28]	Endpoints and relay server	No	Relay or session service	Streaming with 20 to 30 ms budgets	Tens of ms	Mouth to ear, one-way network	Network path, 5G slices, PLC	No
Edge and MEC rendering [12, 4, 20]	Edge server	Partly, one render stage	One rendering service	Real-time streaming	Reported on MEC [12]	End-to-end on the edge	Static MEC placement	No
Fog orchestration of audio scenes [35, 36]	Edge devices	Yes, scene model	Device level services	Soft real-time	Not reported	Not reported	Orchestrator assigns scene elements	No
Audio microservices and DSPaaS [10, 11, 7, 6]	Cloud	No, request pipelines	Task-level services behind a gateway, one server in DSPaaS	Offline, with reactive and low-latency tiers in DSPaaS	0.8 to 1.3 s [10], tiers down to 10 to 30 ms [11]	Request round trip at the gateway, action to sound	Replicas, containerization planned [10]	Partial, VST plugin as client [10]
Cloud native media pipelines [13, 37, 5]	Containers on-premises or cloud	Yes, GStreamer graph	Pipeline per pod with sidecars	Frame synchronous	293 ms, 1.8 s recovery [13]	Command-to-output frame, failure to recover	Restart policies, self-healing, blanker	No
HPC and GPU audio [38, 9]	Cluster or GPU host	Yes, GPU task graph	Not a microservice	Real-time on one host	Milliseconds	Audio callback on the host	Vertical, accelerator	Yes, DAW API [9]
Generic DAG engines [14, 39, 18, 15, 40]	Cloud and edge	Yes, workflow DAG	Function or operator per node	Best effort, no sample deadlines	Broker bound, cold starts	Broker end-to-end, function invocation	Autoscaling, retries, replay	No

shows that these systems have not been measured against the per-block budget, not that they are unable to meet it. This observation answers RQ3.

Microservice principles applied to DSP DAG execution

Microservice principles were formulated for business applications with request and response interactions. **Table 2** maps the main principles to the requirements of a DSP DAG and states how the reviewed works address each principle. Each principle is classified in the text as conflicting, neutral, or requiring an audio-specific interpretation. The mapping answers RQ2.

Table 2. **Microservice principles and their meaning for the execution of DSP DAG graphs**

Microservice principle	Expectation in classic services	Requirement of a DSP DAG	Status in the reviewed work
Single responsibility	One business capability per service	One DSP node type or one subgraph per service, with the plugin host as the unit	Task-level services in [10], no node-level services
Independent deployment	Rolling updates without downtime	Updates must not change a running graph, node versions pinned per session	Not addressed
Statelessness	State in external stores, any replica serves any request	Filters, delays and reverbs hold sample state that must follow the stream	Session affinity implicit in [13], stateful operators in [14]
API contract	REST or gRPC request and response	Block streams with timestamps, ordering, and back pressure	Stream transport between containers only in [13]
Horizontal scaling	Add replicas behind a load balancer	Scale by the number of concurrent graphs, critical path is not shortened	Not tested for DSP graphs in [10]
Resilience patterns	Retries, timeouts, circuit breakers	Retries add latency, dropouts must be concealed within a block period	Self-healing in seconds [13], packet loss concealment in NMP [1]
Observability	Logs, metrics and traces per request	Per-block latency, missed deadlines, clock drift per session	Asynchronous telemetry in [13]
Orchestration	Kubernetes scheduling and autoscaling	Deadline-aware placement, co-location of chained nodes, warm instances	Real-time DAG scheduling theory only [18], cold starts as an open problem [15]

Three principles conflict with DSP DAG execution directly. Statelessness conflicts with the internal state of DSP nodes. A replica cannot take over a filter without receiving its state, and the state changes every block. The practical answer is session affinity, where a graph or a subgraph is pinned to a service instance for the whole session, and failover is treated as a session migration with a short concealment. The request and response API contract conflicts with the streaming nature of audio blocks. A node service must expose a stream endpoint with sequence numbers, timestamps, and back pressure, and the gateway must not sit on the audio path. Resilience patterns such as retries and circuit breakers conflict with the block deadline. A retry that takes longer than one block period is a dropout. Instead, the graph must include concealment nodes, and the orchestrator must keep warm replicas so that failover fits into the latency budget.

Two principles are neutral and can be applied directly. Single responsibility maps to one service per DSP node type or per subgraph, which is the plugin host in a DAW-originated graph. Independent deployment maps to versioned node images that are pinned

per session, so that a rolling update of a node type does not change the sound of a running session. Three principles need an audio-specific interpretation. Horizontal scaling adds capacity for more concurrent graphs but does not reduce the latency of one graph, which is bounded by its critical path. Observability may be extended from request traces to block-level metrics such as missed deadlines and clock drift. Orchestration must become deadline-aware, so that chained nodes are co-located, and remote edges are placed only where the latency budget allows.

The choice of service granularity deserves a separate remark. One service per DSP node gives the finest reuse and isolation but adds a network hop for every edge, which is the worst case for the critical path. One service per subgraph, for example per track chain or per bus, keeps the hops on the critical path small and still allows independent scaling of heavy subgraphs such as convolution reverbs. One service per whole graph is a monolith per session and gives the lowest latency but no reuse of hot node types across sessions. The reviewed works do not evaluate this trade-off. The partitioning heuristics from generic DAG engines and the co-location rule from the media pipeline work are the starting points for such an evaluation [14, 13].

Open problems and what is missing

The review answers the fourth research question with the following list of missing elements.

- Portable representation of DAW graphs. There is no open, engine-independent serialization of a DAW session as a DSP DAG with plugin identifiers, parameter automation, routing, and latency compensation. Web Audio Modules cover the browser [16, 2], GPU APIs cover one host [9], and the VST bridge covers only the user interface [10].
- Granularity model for stateful DSP nodes. No reviewed work defines how to partition a DSP DAG into services, how to pin subgraphs to instances, and how to migrate node state during failover within the latency budget.
- Deadline-aware orchestration. Container orchestrators schedule pods by resources, not by block deadlines. Real-time DAG scheduling theory does not include network hops [18], and serverless platforms suffer from cold starts [15].
- Low-latency block transport between services. Media pipelines keep raw traffic inside a shared namespace [13], and NMP systems optimize the network path [26, 27], but there is no service-to-service audio transport with timestamps, back pressure, and concealment that fits a microservice mesh.
- Concealment and clock discipline as graph elements. Dropouts caused by orchestration actions must be concealed in the graph, and all services of a session must share a clock. These functions are treated as network concerns in NMP and are absent in microservice works.
- Benchmarks and evaluation methodology. Reported latencies are not comparable across families. A benchmark with defined graphs, block sizes, topologies, and measurement points along the critical path is missing, similar to what exists for stream processing microservices [14]. The latency and reliability indicators of DSPaaS are a starting point for such a benchmark [11], and open multitrack material could come from platforms such as Hi-Audio [17].
- Plugin hosting in containers. Licensing, sandboxing, binary compatibility, and GPU access for desktop plugin formats inside containers are not studied, and the container security survey of Jarkas et al. shows that privileged hardware access widens the attack surface [41]. The compute continuum idea suggests a split between latency-critical nodes near the client and heavy nodes in the cluster [38], but no implementation exists.
- Cost and energy models. The elastic cloud model charges per resource time, whereas a session pins resources for its whole duration. The trade-off between per-session monoliths and shared node services is not quantified. Lostanlen and Bouchet warn that rebound effects make digital audio unsustainable when more

compute is offered without accounting for its energy [42], and Gabrielli, Principi and Turchet report case studies on sustainability in the Internet of Sounds [43].

Ranked by the number of included studies that touch each gap, low-latency block transport comes first with 10 studies, followed by the portable representation with 7, concealment and clock discipline and benchmarks with 5 each, deadline-aware orchestration with 4, plugin hosting with 3, cost models with 2, and the granularity model with none. The gap with no study is the one that a microservice-based engine cannot avoid. These gaps are connected. The representation defines the graph, the granularity model defines the services, the transport and the clock connect them, the orchestrator places them, and the benchmark evaluates the result. Progress on one element without the others will not produce a usable cloud audio engine. This is why the next step of our research is a reference architecture that covers all elements and a prototype that executes DAW-originated DSP DAGs as microservices with a measured latency budget.

Answers to the research questions

RQ1. Eight solution families process or render audio as graphs. Browser DAWs, HPC and GPU systems, media pipelines and generic DAG engines use an explicit graph. NMP, edge and fog systems, audio microservices and the DSPaaS vision do not. Only browser DAWs and the GPU API originate from a DAW. The computation model is block-synchronous dataflow on the client and on single hosts, streaming with a budget in NMP and edge systems, and best-effort workflow execution in the cloud.

RQ2. Three principles conflict with block-synchronous execution. Statelessness, request-based API contracts, and retry-based resilience. Workarounds exist in the literature for two of them: session affinity and stream transport inside a shared namespace [13], and packet loss concealment from NMP [30]. Two principles apply without change: single responsibility and independent deployment. Three principles need an audio-specific interpretation: horizontal scaling, observability, and orchestration.

RQ3. The families report latencies at five different points. Browser DAWs report audio buffer delays on the client. NMP systems report mouth-to-ear or one-way network delays. Edge systems report motion-to-sound delays. Audio microservices report request round trips of 0.8 to 1.3 s. Media pipelines report command-to-output delays of 293 ms and recovery times of about 2 s, and serverless platforms add cold starts on top of every invocation. The figures are not comparable across families because they describe different quantities. The cloud and container figures lie two orders of magnitude above the 20 to 30 ms live budget, but they measure a request or a command rather than a single audio block, so they indicate that these systems have not been evaluated on the per-block path rather than that they cannot meet it.

RQ4. Eight elements are missing. Ranked by evidence, they are block transport, graph representation, concealment, and clock discipline, together with benchmarks, deadline-aware orchestration, plugin hosting, cost models, and the service granularity model, which no included study addresses.

Validity challenges

Construct validity. The eight families and the seven criteria were defined by the authors. A different coding could merge the edge and fog families or split the generic DAG engines. The comparison of reported latencies is qualitative, because the papers measure at different points and no reviewed paper reports the transfer delay between two DSP nodes for audio blocks.

Internal validity. The screening and the extraction were performed by one reviewer. The Stage 2 record count is approximate because the publisher sites were searched through a general search engine and not through a database interface with an exportable log. The reported latencies were taken from the papers without re-measurement, and their measurement points differ, which is why **Table 1** records the measurement point next to each figure instead of comparing the figures directly.

External validity. 17 of the 43 included studies come from two proceedings volumes of one symposium, and a large share of them originates from a small number of European research groups. The full-text criterion excluded some work on DAG scheduling and serverless workflows, and 25 topically relevant papers, including edge rendering, fog orchestration, and audio microservice work that would have strengthened the three smallest families. Cloud gaming, WebRTC, and broadcast transport standards were covered by a small number of studies. The findings should be read as a characterization of the peer-reviewed literature in the selected venues, not of commercial practice.

Conclusion validity. The gap ranking counts citations, not the depth of treatment, and the counts are small. The statement that cloud and container latencies exceed the live budget by two orders of magnitude rests on a handful of reported figures from two papers. It is robust to the choice of block size, because a block of 256 or 512 samples changes the budget arithmetic by a factor of two or four while the reported figures exceed the budget by a factor of one hundred or more.

CONCLUSION

This paper reviewed existing solutions for processing and rendering audio in the cloud as DSP DAG graphs with a focus on microservice architecture. 43 peer-reviewed studies were selected with an explicit protocol and grouped into eight solution families that range from browser DAWs, which execute an explicit DAW-originated graph on the client, to generic DAG engines, which execute graphs in the cloud without an audio deadline. The families in between move streams, single rendering stages, offline tasks, or frame-level media graphs to the edge or the cloud.

The comparison shows that no reviewed system combines an explicit DSP DAG, a DAW origin, block-level real-time execution, and a microservice deployment. The mapping of microservice principles to DSP DAG requirements shows that statelessness, request-based API contracts, and retry-based resilience conflict with block-synchronous audio, whereas single responsibility and independent deployment apply directly, and scaling, observability, and orchestration need an audio-specific interpretation. The comparison of reported latencies shows that the reviewed studies report different types of latency. Browser DAWs report the audio buffer delay on the client, networked music performance systems report the one-way network or mouth-to-ear delay, edge systems report the motion-to-sound delay, audio microservices report the request-response time at the gateway, and media pipelines report the command-to-output delay. These quantities are measured on different paths and with different workloads, so direct comparisons between families are difficult. The request-response and command-to-output figures of the cloud and container systems, from hundreds of milliseconds to seconds, lie two orders of magnitude above the 20 to 30 ms budget of networked music performance, but they describe the processing of a whole request or command and not the per-block path that this budget constrains. The figures therefore do not show that the existing systems fail to meet real-time audio requirements. They show that no reviewed cloud or container system has been evaluated against a per-block latency budget. A latency model that makes the reported figures comparable and a measurement of the transfer cost between chained node services are the subject of a companion article.

What such systems lack today is a portable representation of DAW graphs, a granularity model for stateful DSP nodes, deadline-aware orchestration, a low-latency block transport between services, concealment and clock discipline as graph elements, a common benchmark, plugin hosting in containers, and a cost model. Further research will address these elements with a reference architecture for microservice-based execution of DAW-originated DSP DAGs and with a prototype evaluated against the latency budgets established by networked music performance research.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors express their gratitude to the Ministry of Education and Science of Ukraine for support (Project No 0125U001883).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [O.C.]; methodology, [O.C., H.K.]; validation, [O.C., H.K.]; formal analysis, [O.C.]; investigation, [O.C.]; resources, [O.C., H.K.]; data curation, [O.C.]; writing, original draft preparation, [O.C.]; writing, review and editing, [H.K.]; visualization, [O.C.]; supervision, [H.K.]; project administration, [H.K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Turchet, L., Lagrange, M., Rottondi, C., Fazekas, G., Peters, N., Østergaard, J., Font, F., Bäckström, T., & Fischione, C. (2023). The Internet of Sounds: Convergent trends, insights and future directions. *IEEE Internet of Things Journal*, *10*(13), 11264–11292. https://doi.org/10.1109/JIOT.2023.3253602

[2] Buffa, M., & Demont, S. (2024). Can you DAW it online? The challenges of a web-based open source workstation. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–8). IEEE. https://doi.org/10.1109/IS262782.2024.10704163

[3] Dewey, C., Moore, A., & Lee, H. (2024). Practitioners' perspectives on spatial audio: Insights into Dolby Atmos and binaural mixes in popular music. *Journal of the Audio Engineering Society*, *72*(7/8), 504–516. https://doi.org/10.17743/jaes.2022.0153

[4] Giacomelli, S., Centofanti, C., Santos, J., Galbiati, M., Salvi, T., Graziosi, F., & Rinaldi, C. (2024). Remote immersive audio production: State of the art implementation, challenges, and improvements. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–10). IEEE. https://doi.org/10.1109/IS262782.2024.10704192

[5] Fukudome, D., & Nishimura, S. (2024). Cloud-based Internet linear video delivery system using edge computing on a CDN. *SMPTE Motion Imaging Journal*, *133*(6). https://doi.org/10.5594/JMI.2024/TPSA1162

[6] Chaplia, O., Klym, H., Popov, A. I., & Konuhova, M. (2025). Enhancing REST API handlers organization for Node.js microservices. *SN Computer Science*, *6*, Article 776. https://doi.org/10.1007/s42979-025-04311-8

[7] Chaplia, O., & Klym, H. (2023). Node.js framework for automated code scaling and execution on multiple distributed machines. In *2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)*. IEEE. https://doi.org/10.1109/CSIT61576.2023.10324052

[8] Turchet, L., & Rottondi, C. (2023). On the relation between the fields of Networked Music Performances, Ubiquitous Music, and Internet of Musical Things. *Personal and Ubiquitous Computing*, *27*(5), 1783–1792. https://doi.org/10.1007/s00779-022-01691-z

[9] Bianchi, D., Avanzini, F., Baratè, A., Ludovico, L. A., & Presti, G. (2022). A GPU-oriented application programming interface for digital audio workstations. *IEEE Transactions on Parallel and Distributed Systems*, *33*(8), 1924–1938. https://doi.org/10.1109/TPDS.2021.3131659

[10] Raposo, A. N., & Soares, V. N. G. J. (2025). Modular microservices architecture for generative music integration in digital audio workstations via VST plugin. *Future Internet*, *17*(10), Article 469. https://doi.org/10.3390/fi17100469

[11] Turchet, L., & Krstulović, S. (2024). DSP as a service: Foundations and directions. *IEEE Open Journal of the Communications Society*, *5*, 6212–6226. https://doi.org/10.1109/OJCOMS.2024.3464696

[12] Martusciello, F., Centofanti, C., Rinaldi, C., & Marotta, A. (2023). Edge-enabled spatial audio service: Implementation and performance analysis on a MEC 5G infrastructure. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–8). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335480
[13] Herranz Sánchez, M., Llorente, Á., del Rio, A., & Jimenez, D. (2026). Beyond hardware mixers: A resilient cloud-native architecture for real-time remote video production. *Applied Sciences*, *16*(18), Article 9253. https://doi.org/10.3390/app16189253
[14] Henning, S., & Hasselbring, W. (2024). Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud. *Journal of Systems and Software*, *208*, Article 111879. https://doi.org/10.1016/j.jss.2023.111879
[15] Raith, P., Nastic, S., & Dustdar, S. (2023). Serverless edge computing—Where we are and what lies ahead. *IEEE Internet Computing*, *27*(3), 50–64. https://doi.org/10.1109/MIC.2023.3260939
[16] Buffa, M., & Vidal-Mazuy, A. (2023). WAM-studio, a digital audio workstation (DAW) for the Web. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion)* (pp. 543–548). ACM. https://doi.org/10.1145/3543873.3587987
[17] Gil Panal, J. M., David, A., & Richard, G. (2026). The Hi-Audio online platform for recording and distributing multi-track music datasets. *EURASIP Journal on Audio, Speech, and Music Processing*, *2026*, Article 26. https://doi.org/10.1186/s13636-026-00459-0
[18] Verucchi, M., Olmedo, I. S., & Bertogna, M. (2023). A survey on real-time DAG scheduling, revisiting the Global-Partitioned Infinity War. *Real-Time Systems*. Advance online publication. https://doi.org/10.1007/s11241-023-09403-3
[19] Turchet, L., & Antoniazzi, F. (2023). Semantic Web of Musical Things: Achieving interoperability in the Internet of Musical Things. *Journal of Web Semantics*, *75*, Article 100758. https://doi.org/10.1016/j.websem.2022.100758
[20] Laitinen, M.-V., Vasilache, A., Rämö, A., Paulus, J., & Eksler, V. (2025). Combined object-based audio and MASA format for enhanced spatial mobile communication. In *ICASSP 2025, 2025 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49660.2025.10889750
[21] Ackva, V., & Schulz, F. (2024). ANIRA: An architecture for neural network inference in real-time audio applications. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–10). IEEE. https://doi.org/10.1109/IS262782.2024.10704099
[22] Matuszewski, B., & Golvet, A. (2023). Rapid prototyping of distributed musical things using web technologies. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–5). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335368
[23] Buffa, M., Girard, D., & Hofr, A. (2024). Using Web Audio Modules for immersive audio collaboration in the Musical Metaverse. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–10). IEEE. https://doi.org/10.1109/IS262782.2024.10704108
[24] Turchet, L., & De Cet, M. (2023). A web-based distributed system for integrating mobile music in choral performance. *Personal and Ubiquitous Computing*, *27*(5), 1829–1842. https://doi.org/10.1007/s00779-023-01709-0
[25] Boem, A., Tomasetti, M., & Turchet, L. (2025). Between immersion and usability: A comparative study of 2D and mixed reality interfaces for remote music making. *International Journal of Human-Computer Studies*, *203*, Article 103586. https://doi.org/10.1016/j.ijhcs.2025.103586
[26] Turchet, L., & Tomasetti, M. (2023). Immersive networked music performance systems: Identifying latency factors. In *2023 Immersive and 3D Audio: From Architecture to Automotive (I3DA)*. IEEE. https://doi.org/10.1109/I3DA57090.2023.10289169
[27] Turchet, L., & Casari, P. (2023). Assessing a private 5G SA and a public 5G NSA architecture for networked music performances. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–6). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335359
[28] Turchet, L., & Casari, P. (2024). The Internet of Musical Things meets satellites: Evaluating Starlink support for networked music performances in rural areas. In *2024*

IEEE 5th International Symposium on the Internet of Sounds (IS2) (pp. 1–8). IEEE. https://doi.org/10.1109/IS262782.2024.10704207
[29] Tomasetti, M., & Turchet, L. (2023). Playing with others using headphones: Musicians prefer binaural audio with head tracking over stereo. *IEEE Transactions on Human-Machine Systems*, *53*(3), 501–511. https://doi.org/10.1109/THMS.2023.3270703
[30] Vignati, L., Dallona, S., & Turchet, L. (2023). PLC Testbench: A modular tool for the study and comparison of audio packet loss concealment algorithms. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–8). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335357
[31] Bert, D., Domini, N., Peloso, R., Severi, L., Sacchetto, M., Bianco, A., & Rottondi, C. (2023). FPGA-based low-latency audio coprocessor for networked music performance. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–8). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335333
[32] Cochard, P., Weber, J., Michon, R., Risset, T., & Letz, S. (2024). Ethernet real-time audio transmission to FPGA. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–7). IEEE. https://doi.org/10.1109/IS262782.2024.10704104
[33] Dürre, J., & Werner, N. (2024). DECT-2020 NR for professional live audio: Design space exploration and practical evaluation. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–7). IEEE. https://doi.org/10.1109/IS262782.2024.10704159
[34] Stefani, D., & Turchet, L. (2023). Real-time embedded deep learning on Elk Audio OS. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–10). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335204
[35] Mikolajczyk, K., Trolland, S., Ilsar, A., McCormack, J., Ferguson, S., & Bown, O. (2023). Gestural interactions with object-based audio in an Internet of Sounds ecosystem. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–9). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335408
[36] Vieira, R., Wei, S., Röggla, T., Muchaluat-Saade, D. C., & Cesar, P. (2024). Immersive Io3MT environments: Design guidelines, use cases and future directions. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–10). IEEE. https://doi.org/10.1109/IS262782.2024.10704141
[37] del Rio, A., Llorente, Á., Ortiz-Arce, S., Belesioti, M., Pappas, G., Muñiz, A., Contreras, L. M., & Christopoulos, D. (2025). Seamless user-generated content processing for smart media: Delivering QoE-aware live media with YOLO-based bib number recognition. *Electronics*, *14*(20), Article 4115. https://doi.org/10.3390/electronics14204115
[38] Turchet, L., Vella, F., & Fiore, S. (2023). The potential of high-performance computing for the Internet of Sounds. In *2023 4th International Symposium on the Internet of Sounds* (pp. 1–7). IEEE. https://doi.org/10.1109/IEEECONF59510.2023.10335270
[39] Fragkoulis, M., Carbone, P., Kalavri, V., & Katsifodimos, A. (2024). A survey on the evolution of stream processing systems. *The VLDB Journal*, *33*, 507–541. https://doi.org/10.1007/s00778-023-00819-8
[40] Dakić, V., Kovač, M., & Slovinac, J. (2024). Evolving high-performance computing data centers with Kubernetes, performance analysis, and dynamic workload placement based on machine learning scheduling. *Electronics*, *13*(13), Article 2651. https://doi.org/10.3390/electronics13132651
[41] Jarkas, O., Ko, R. K. L., Dong, N., & Mahmud, R. (2025). A container security survey: Exploits, attacks, and defenses. *ACM Computing Surveys*, *57*, 1–36. https://doi.org/10.1145/3715001
[42] Lostanlen, V., & Bouchet, L. (2024). Rebound effects make digital audio unsustainable. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)* (pp. 1–7). IEEE. https://doi.org/10.1109/IS262782.2024.10704126
[43] Gabrielli, L., Principi, E., & Turchet, L. (2025). Sustainability and the Internet of Sounds: Case studies. *IEEE Transactions on Technology and Society*, *6*(2), 165–180. https://doi.org/10.1109/TTS.2024.3513777

ОГЛЯД МЕТОДІВ ХМАРНОГО ОПРАЦЮВАННЯ ЗВУКУ НА ОСНОВІ ОРІЄНТОВАНИХ АЦИКЛІЧНИХ ГРАФІВ ОБРОБЛЕННЯ СИГНАЛІВ У МІКРОСЕРВІСНИХ АРХІТЕКТУРАХ

Олег Чапля, Галина Клим*
Національний університет "Львівська політехніка",
вул. Степана Бандери, 12, Львів, 79013, Україна

АНОТАЦІЯ

Вступ. Ланцюги обробки аудіо у цифрових звукових робочих станціях (ЦЗРС, англ. DAW – digital audio workstations) є орієнтованими ациклічними графами (ОАГ, англ. DAG – directed acyclic graph) вузлів цифрового оброблення сигналів (ЦОС, англ. DSP – digital signal processing). Виконання таких графів у хмарних середовищах дає змогу забезпечити гнучкість обчислювальних ресурсів, можливість спільної роботи та доступ із термінальних пристроїв. Мікросервісна архітектура переважає у хмарних системах, але її принципи сформовані навантаженнями типу запит-відповідь, а не блоковою синхронною обробкою аудіо. У статті оглянуто наявні методи для хмарного оброблення та відтворення звуку, поданого у вигляді графів цифрового оброблення сигналів, та оцінено їх на відповідність принципам мікросервісного підходу.

Матеріали та методи. Застосовано систематичний протокол пошуку з явними критеріями включення до двох повних томів праць IEEE International Symposium on the Internet of Sounds, до пошуку в базах даних і на сайтах видавців та до зворотного відбору за списками літератури. З 97 праць симпозіуму та 48 повних текстів з інших джерел включено 43 рецензовані дослідження з доступними повними текстами, опубліковані з 2022 до 2026 року. Методи згруповано у вісім сімейств та оцінено за розташуванням, явністю графа, деталізацією сервісів, режимом реального часу, повідомленими затримками, механізмами масштабування та відмовостійкістю, а також підтримкою графів, що походять із цифрових звукових робочих станцій.

Результати. Браузерні ЦЗРС зберігають ОАГ на клієнті та використовують хмарне середовище для зберігання даних і спільної роботи. Системи мережевого музичного виконання ретранслюють потоки у межах жорстких бюджетів затримки, але не виконують офлайн-графів опрацювання звуку віддалено. Периферійні служби формування звуку переносять на периферію лише один етап. Мікросервіси виконують пакетні завдання через мережеві програмні інтерфейси. Хмарні медіаконвеєри виконують обробку звукових сигналів у контейнерах із самовідновленням, а універсальні рушії орієнтованих ациклічних графів планують і масштабують, але не гарантують точності на рівні окремих значень блоків. Жодна з оглянутих систем не виконує граф ЦОС, що походить із ЦЗРС, як набір мікросервісів із граничними термінами на рівні блоків. Затримки в оглянутих матеріалах виміряно в різних точках, тому їх неможливо порівняти. Хмарні та контейнерні системи повідомляють про кількість від сотень мілісекунд до кількох секунд.

Висновки. Наявним системам бракує переносного подання графів ЦЗРС, моделі подрібнення для вузлів цифрової обробки сигналів зі станом, планування за граничними термінами в засобах оркестрування контейнерів, передавання блоків між сервісами з малою затримкою та еталонних тестів хмарного формування звуку. Ці прогалини визначають програму досліджень хмарного опрацювання звуку на основі мікросервісів.

Ключові слова: хмарні обчислення, мікросервіси, цифрова обробка сигналів, орієнтований ациклічний граф, цифрова звукова робоча станція, надійність

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
08 September, 2026	18 September, 2026	21 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 97–108
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 519.246.8:504.064.2

ANALYSIS AND FORECASTING OF RIVER WATER POLLUTION DYNAMICS USING LONG SHORT-TERM MEMORY NEURAL NETWORKS AND MULTIFRACTAL ANALYSIS

***Mykhailo Bordun*[1] & *Yaroslav Sokolovskyy*[2]**
[1] *Information Systems Department, Ivan Franko National University of Lviv, 1 Universytetska Street, Lviv, 79007, UKRAINE*
[2] *Department of Computer-Aided Design, Lviv Polytechnic National University, 5 Metropolyta Andreia Street, Lviv, 79016, UKRAINE*

Bordun, M.I., Sokolovskyy, Y.I. (2026). Analysis and Forecasting of River Water Pollution Dynamics Using Long Short-Term Memory Neural Networks and Multifractal Analysis. *Electronics and Information Technologies*, 35, 97–108. https://doi.org/10.30970/eli.35.7

ABSTRACT

Background. Organic water pollution has complex, nonlinear, nonstationary, and self-similar dynamics that conventional forecasting models often fail to represent. This study develops an automated hybrid framework combining multifractal analysis and deep learning to improve Biochemical Oxygen Demand (BOD) forecasting and support early risk identification.

Materials and Methods. A retrospective BOD time series from a river monitoring station was processed through an automated data and modeling pipeline. Sliding windows extracted localized features and captured long-range correlations. Multifractal Detrended Fluctuation Analysis (MFDFA) estimated generalized Hurst exponents and singularity spectra for each window. A fractional-exponential approximation converted the continuous spectra into discrete coefficients while preserving their structural characteristics. These multifractal descriptors were combined with statistical measures to form the feature space. A Long Short-Term Memory (LSTM) network was trained using time-series cross-validation and hyperparameter search. Results were compared with Autoregressive Integrated Moving Average (ARIMA) and Autoregressive Fractionally Integrated Moving Average (ARFIMA) benchmarks.

Results and Discussion. The hybrid MFDFA-LSTM framework produced more accurate and stable forecasts than the benchmark models across short-term and extended horizons. Fractional-exponential descriptors supplied information about multi-scale fluctuations, persistence, spectrum width, and asymmetry, enabling the LSTM to represent nonlinear pollution dynamics more effectively. Hyperparameter optimization balanced temporal memory, model complexity, and generalization, reducing root mean square error (RMSE) and maintaining stable predictive performance.

Conclusion. The study validates an automated system for water-quality forecasting. Its innovation is transforming continuous multifractal spectra into informative discrete features through fractional-exponential approximation. Combined with LSTM modeling, this framework improves forecast reliability and provides environmental authorities with an early-warning and decision-support tool for proactive water-quality management.

Keywords: LSTM, BOD, multifractal analysis, MFDFA, ARFIMA, singularity spectrum.

INTRODUCTION

The aquatic environment is currently undergoing significant transformation driven by the dual pressures of global climate change and the rapid expansion of industrial and agricultural

© 2026 Mykhailo Bordun & Yaroslav Sokolovskyy. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

sectors [1, 2]. Statistical data indicates a staggering global discharge of approximately 400 billion cubic meters of wastewater into natural water bodies, leading to the contamination of nearly 5.5 trillion cubic meters of water worldwide [3]. Critically, over 80% of this wastewater enters the environment without adequate treatment. This escalating anthropogenic burden systematically erodes the self-restoration capacity of aquatic ecosystems [4].

The Dnipro River is fundamental to Ukraine's water security, supplying more than 70% of the population and driving the country’s industrial and agricultural productivity. However, the basin is currently under immense anthropogenic stress. Among the pollutants recorded by the monitoring network, organic substances pose a significant threat to the ecosystem’s health [5]. The resulting depletion of dissolved oxygen, combined with nutrient loading from urban and agricultural sources, fosters massive water 'blooming' through the intensive reproduction of blue-green algae.

The diversity of contaminants infiltrating river systems underscores the absence of a singular, universal monitoring methodology. Effective ecosystem management necessitates continuous, real-time tracking of various physicochemical parameters to ensure compliance with environmental standards. Among these, Biochemical Oxygen Demand (BOD) serves as a critical integrative indicator of water quality. It quantifies the concentration of dissolved oxygen consumed by aerobic microorganisms during the biochemical oxidation of organic matter into stable inorganic compounds [6]. Consequently, BOD provides a comprehensive assessment of the biological load on the aquatic environment. Elevated BOD levels signify excessive contamination from municipal or agricultural discharges, posing a severe risk of hypoxia and significantly impeding the river's natural self-purification mechanisms.

Notwithstanding the rapid advancements in electronic and sensor technologies, the direct real-time measurement of BOD via physicochemical sensors remains unattainable. Consequently, traditional laboratory methods are inherently incapable of providing the early-warning functions necessary for environmental disaster prevention. To address this limitation, environmental monitoring practices are increasingly shifting toward predictive strategies. By leveraging easily measurable parameters — such as temperature, pH, electrical conductivity, dissolved oxygen, and turbidity-mathematical models can 'calculate' probable BOD concentrations. This approach underpins the 'soft sensor' concept: a software-based algorithm that acts as a virtual replacement for complex laboratory analyses [7, 8]. This paradigm necessitates the application of neural networks, which are uniquely suited to identifying the intricate, nonlinear correlations between these proxy indicators. Alternatively, predictive modeling can be driven by longitudinal datasets of historical BOD observations [9-12]. The precision of such forecasts is directly correlated with the scale and quality of the historical database.

While standard methods, such as Support Vector Regression (SVR), provide reliable point forecasts, they often fail to capture the underlying complexity of dynamic systems. This limitation stems from two primary factors: first, the 'inertial memory' inherent in the Dnipro River system, where current water quality is intrinsically linked to upstream historical events — a dependency that classical algorithms often treat as a series of independent occurrences. Second, the physicochemical parameters within the Dnipro basin exhibit intricate nonlinear correlations; for instance, the relationship between dissolved oxygen and BOD is highly non-stationary and fluctuates with seasonal temperature shifts. Building upon the findings of recent research [13], which highlights the potential of hybrid modeling for such complex systems, this article aims to develop a synergistic framework for BOD prediction. This approach integrates a Long Short-Term Memory (LSTM) recurrent architecture, capable of retaining systemic temporal dependencies, with multifractal analysis for advanced data characterization and preprocessing.

MATERIALS AND METHODS

Software tools analysis

The computational framework for this study integrates both Python and R to leverage their respective analytical strengths. Data preprocessing and the implementation of the LSTM architecture were performed in Python, utilizing TensorFlow for deep learning and NumPy and Pandas for efficient numerical operations. However, the multifractal characterization and fractal modeling using the AutoRegressive Fractionally Integrated Moving Average (ARFIMA) were executed in the R environment. This dual-language approach was adopted because R offers a more mature and comprehensive ecosystem for fractal analysis, specifically through the mfdfa, forecast, and arfima packages, which currently lack equivalent robust implementations in Python's standard libraries.

Dataset of river water pollution

Data utilized in this research originates from the official environmental monitoring program of Ukraine, accessed via data.gov.ua [14]. The dataset encompasses a wide array of water quality indicators collected from 2003 onwards across major river basins. To ensure the high-frequency and long-term data needed for multifractal modeling, this study focuses on the Dnipro-Donbas Canal station. This site was selected because it provides the largest dataset, containing 410 observations. While the monitoring station records various indicators such as nitrogen, ammonium ions and dissolved oxygen levels, BOD was specifically isolated as the critical variable for assessing organic pollution due to its comprehensive ability to reflect the oxygen-consuming potential of organic matter.

Raw data were retrieved using automated HTTP GET protocols, yielding numerous CSV-formatted files covering various time intervals. The data extraction focused exclusively on monitoring records from the Dnipro-Donbas Canal station. Following retrieval, the dataset underwent preprocessing to address non-recorded BOD measurements, thereby validating its suitability for deep learning applications. The final, refined time series was structured to include only chronological date records and their corresponding BOD values.

Investigating seasonality, trend, stationarity, and long-memory presence

To ensure the best model selection, we first analyzed the characteristics of the Dnipro River BOD time series. Following data cleaning and preparation, we studied the measurements' statistical properties to build a reliable foundation for the forecasting process.

Fig. 1 depicts the time series decomposition, highlighting well-defined seasonal fluctuations and an overall trend. The remaining residual series captures the noise component, providing a clear visualization of data outliers that deviate from the established patterns. The residual analysis reveals a high degree of model convergence over the stable period. However, distinct spikes in the residual plot at the temporal boundaries suggest transient non-linear shocks in the BOD levels that exceed the predictive capacity of the trained model. These anomalies highlight the stochastic nature of the river ecosystem.

An important characteristic of time series analysis is examining evidence of long-term memory. **Fig. 2** presents the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) graphs. The ACF measures the linear relationship between the current value of the series and its past values at specific lags, capturing both direct and indirect effects. In contrast, the PACF isolates the direct correlation between an observation and its lag by removing the influence of intermediate observations. In the analyzed dataset, several time lags in both the ACF and PACF plots extend beyond the 95% confidence interval (depicted by the shaded blue area). This observation confirms that the series possesses a structured memory, which carries critical implications for model selection.

To validate the stationarity of the Dnipro River BOD data, we applied the ADF and KPSS statistical tests. The ADF test rejected the null hypothesis of non-stationarity

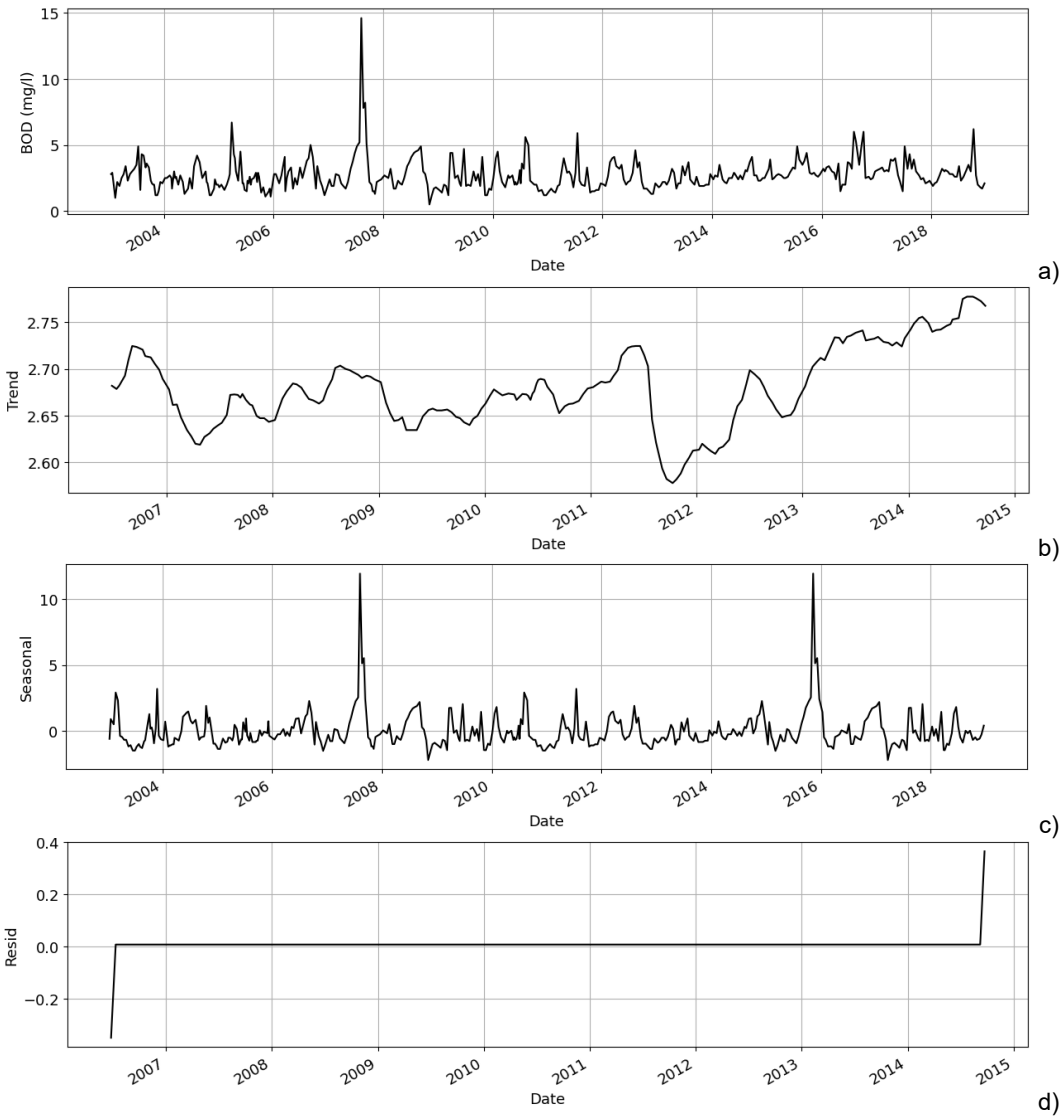


Fig. 1. BOD time series of the Dnipro River (a), decomposed trend (b), seasonality (c) and residuals (d)

($p < 0.05$), while the KPSS test failed to reject the null hypothesis of stationarity ($p > 0.05$) [15]. The consistency between these two complementary tests confirms that the time series does not possess a unit root and is characterized by stationary dynamics, facilitating more accurate modeling.

Mathematical description of the MFDFA method

While sharing core conceptual mechanics with conventional Detrended Fluctuation Analysis (DFA), Multifractal Detrended Fluctuation Analysis (MFDFA) [16] introduces expanded evaluation phases designed to analyze a diverse spectrum of scaling exponents. Consequently, this technique has become a standard analytical instrument for uncovering hidden non-linear structures and determining the degree of multifractality in temporal data.

The execution of the MFDFA algorithm yields a distinct set of mathematical outputs that characterize the multi-scale structure of the time series. The key mathematical vectors output by this pipeline are the generalized Hurst exponent curve (h(q)), the scaling exponent function (τ(q)), and the singularity spectrum (f(α)). This spectrum plots the

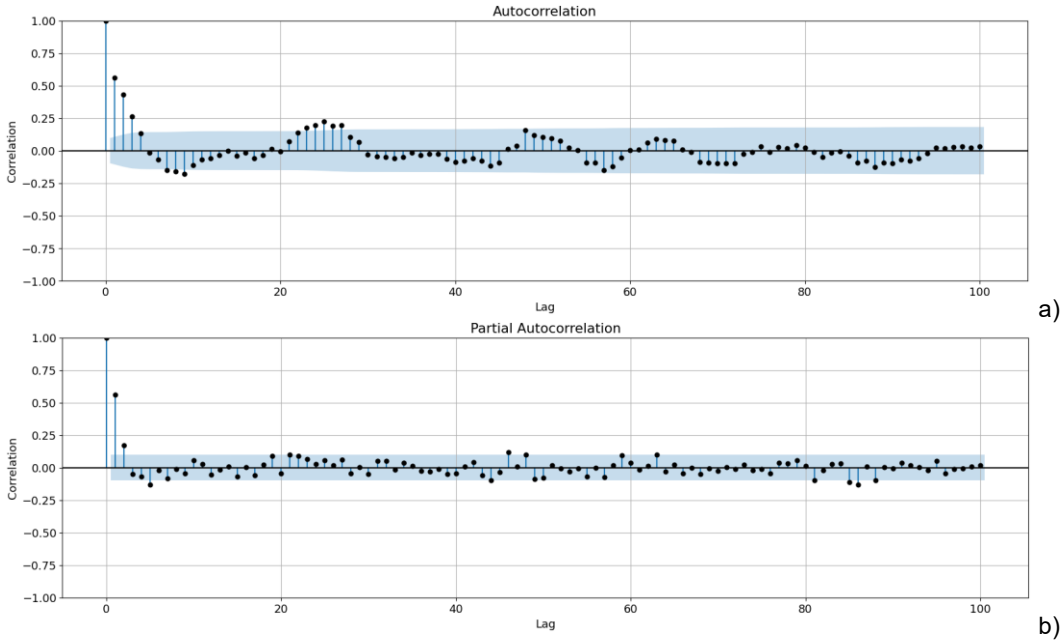


Fig. 2. ACF (a) and PACF (b) functions for the BOD time series of the Dnipro River

singularity dimension against the singularity strength α, providing the foundational curves necessary for calculating crucial structural parameters such as the spectrum width, asymmetry, and peak location.

The mathematical apparatus for the MFDFA processing and the subsequent singularity spectrum f(α)) derivation follows the exact theoretical framework detailed in [12, 16].

To derive quantitative descriptors, the multifractal spectrum is approximated by a second-order polynomial centered at its maximum α_0 by the following function:

$$f(\alpha) = A(\alpha - \alpha_0)^2 - B(\alpha - \alpha_0) + C, \quad (1)$$

where the coefficients A, B, and C are obtained by the least squares method [16].

To provide a more versatile and adaptable fit for the multifractal spectrum, an exponential function was proposed as follows:

$$f(\alpha) = A(\alpha - \alpha_{min})^{\beta_1}(\alpha_{max} - \alpha)^{\beta_2} + \varepsilon, \quad (2)$$

where the coefficients A, β_1, β_2 and ε are obtained by the nonlinear least squares method.

RESULTS AND DISCUSSION

The raw time series was organized into sliding windows of 150 observations each, advancing by one point at a time. This window size provided enough observations for reliable multifractal and statistical analysis while preserving sufficient temporal resolution. This approach allowed us to calculate multifractal and statistical descriptors for every step. We chose this specific window size to balance two needs: providing enough data for an accurate multifractal analysis and creating a long enough sequence to effectively train the LSTM model.

To quantify the multifractal properties of each data window, we employed the MFDFA algorithm, utilizing the specialized MFDFA R package. Key parameters included a defined scale range for segment-wise detrending and a comprehensive set of q-values.

The scaling range for the MFDFA procedure was optimized based on the temporal span of the dataset to ensure a robust and statistically reliable detrending process. We defined the minimum scale at 5 data points ($s_{min} = 5$) to maintain accuracy in the local fluctuation calculation. The maximum scale (s_{max}) was dynamically set at 25% of the window length (N/4), following standard recommendations to avoid artifacts caused by insufficient segment counts at large scales. To achieve a uniform distribution across all time scales, the values were calculated using an exponential sequence of 25 unique points, determined by:

$$s = \exp\left[\ln(s_{min}) + i * \frac{\ln(s_{max}) - \ln(s_{min})}{n - 1}\right], \quad (3)$$

where i ranges from 0 to n – 1, and n = 25.

The q parameter, ranging from –10 to 10, served as a multi-resolution filter: negative values were used to capture the scaling behaviour of small fluctuations, while positive values were dedicated to assessing the scaling of large-magnitude events, thereby providing a complete multifractal profile of the river's organic load.

The MFDFA procedure generated a comprehensive set of parameters, including the Hurst exponent, scaling exponent, singularity spectrum, and fluctuation function. To characterize each sliding window with discrete multifractal properties, the following values were extracted: the Hurst exponent at q = 2 (H(2)), which describes the persistence and long-range correlations of the signal; ΔH, which reflects the variation of the Hurst exponent across different (q)-orders; the spectrum width $\Delta\alpha$, which quantifies the degree of multifractality; α_0, the singularity exponent corresponding to the maximum of the spectrum; the asymmetry parameter B, and the estimated coefficients A, β_1, β_2 derived from the exponential expression in Eq. (2). Here, A controls the overall amplitude of the approximated spectrum, while β_1 and β_2 determine its shape on the left and right sides, respectively. Their relative values describe how rapidly the spectrum rises and falls and therefore provide additional information about its asymmetry.

A comparison of fitting residuals confirmed that the fractional-exponential model provided a more precise representation of the empirical data. This is evidenced by its lower root mean square error (RMSE) of 0.012, which represents a substantial improvement over the quadratic fit's error of 0.034. The superior performance of this approach in representing the empirical $f(\alpha)$ values were statistically proven in our analysis and visually demonstrated in the comparative plots in **Fig. 3**.

To construct the high-dimensional feature space for the 100-step-ahead forecast, we integrated discrete multifractal properties with fundamental statistical descriptors. This feature vector included volatility, slope, mean, entropy, skewness, kurtosis, and data-window width. The computational extraction was performed using the R-language, with the resulting multidimensional dataset exported as a CSV file to a local data repository. This intermediate storage facilitated a seamless transition to the Python-based environment for the subsequent modeling and forecasting phases.

The original time series contained 410 BOD observations. Applying a 150-observation sliding window produced 261 possible feature windows. However, the last window could not be paired with a subsequent BOD observation; therefore, 260 valid feature–target pairs were available for forecasting. The final 100 pairs were reserved as an independent chronological test set, leaving (N = 160) observations for model development. Thus, (N = 160) refers to the number of labeled features — target samples used for training and validation, rather than the number of raw measurements. The 100-step-ahead forecast should also be understood as a 100-observation forecast horizon composed of consecutive one-step-ahead predictions.

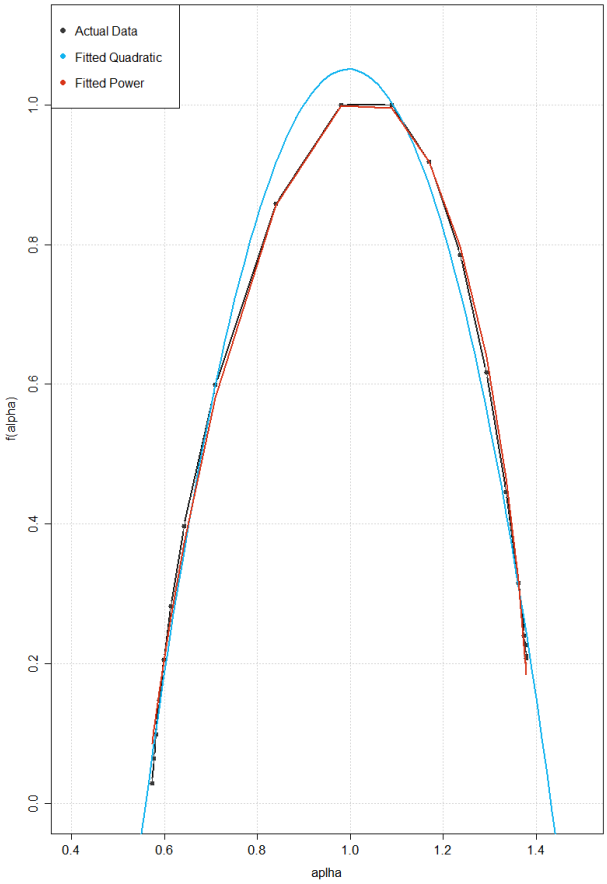


Fig. 3. Singularity spectrum and its quadratic (blue line) and fractional-exponential (red line) approximations

The predictive modeling phase was initiated by aligning each high-dimensional feature vector with its corresponding target BOD observation. We structured the data for supervised learning by aligning the multifractal and statistical features of each 150-observation window with the immediate following BOD record. In other words, the first 150 data points were used to predict the 151st observation, the next shifted window predicted the following observation, and so forth. This one-step-ahead mapping generated a continuous sequence of training pairs. The data were divided chronologically, without random shuffling, to prevent information from future observations from entering the training process. Within the first 160 observations, three expanding time-series folds were used for validation: earlier observations formed the training subset, while the subsequent contiguous observations formed the validation subset. The final 100 observations remained untouched until the final performance evaluation.

The LSTM was selected for its ability to utilize internal memory cells and gating mechanisms, which allow the model to retain long-term dependencies and capture the “inertial memory” inherent in the Dnipro River’s pollution dynamics [17]. The model used a many-to-one structure: several consecutive feature vectors were provided as input, and the network generated one prediction for the immediately following BOD observation. In the final configuration, the input consisted of consecutive time steps and the selected feature variables. A single LSTM layer with hidden units and a hyperbolic-tangent activation function transformed this sequence into a temporal representation. Each LSTM unit maintained both a cell state and a hidden state, while the input, forget, and output gates controlled which information was stored, discarded, or passed to the next time step. The

resulting hidden representation was passed through a dropout layer to reduce overfitting and then through a fully connected output layer with one neuron for the BOD prediction. The network was trained using the Adam optimizer and mean squared error loss.

To ensure the LSTM architecture remained robust despite a relatively small sample size (N = 160), a feature selection process was conducted prior to training. By limiting the input space to an optimal range of 5–7 features, we prevented the model from capturing noise, thereby addressing potential overfitting constraints. Feature and target values were scaled using parameters calculated from the corresponding training data. This dimensionality reduction, together with chronological validation, dropout regularization, and early stopping, was critical for maintaining the reliability of the multifractal-based forecasting.

We employed a tree-based Random Forest Regressor algorithm to perform feature importance estimation. This allowed for a streamlined selection of the most influential predictors, ensuring a more efficient and robust forecasting process by utilizing only the highest-ranking features. The forecast was executed using a refined feature space consisting of six key predictors: width, mean, Hurst exponent, entropy, α_0, and slope. Together, these variables represent the signal's multifractal scaling, complexity, local variability, and temporal trend, providing the LSTM network with a compact but informative description of the BOD dynamics.

A critical challenge in this stage involved optimizing the model's hyperparameters, which define the network's learning capacity and temporal sensitivity [18]. Specifically, we tuned the lookback window, which determines the 'temporal memory' or the number of previous time steps the model analyses to capture the inertial dynamics of the river system. We also optimized the number of model units, representing the hidden neurons that facilitate the learning of complex, non-linear mappings between multifractal descriptors and pollution levels. To control the learning process, the batch size (the number of samples processed before internal weight updates) and the number of epochs (full passes through the training set) were carefully calibrated to ensure convergence without overfitting.

To determine the most robust model configuration, we employed a hyperparameter grid search integrated with time-series cross-validation (a specialized k-fold method for sequential data). We implemented an exhaustive grid search to optimize the neural network, testing combinations of model units (16, 24, 32), batch sizes (8, 16, 24), epochs (100, 150), and lookback windows (2, 3, 4). This rigorous evaluation of 54 distinct configurations allowed for the selection of an optimal model structure. The process was designed to balance high-precision forecasting with robust performance across different stages of the water quality time series.

By utilizing the RMSE metric to minimize error across these chronological splits, the optimal hyperparameter set was selected to produce the final forecasting output. The optimal LSTM configuration for the 100-point forecasting horizon of the Dnipro River BOD time series was identified through a minimized RMSE of 0.8617. This top-performing model utilized a lookback window of 2, a batch size of 8, 100 training epochs, and 32 hidden units. Accordingly, the network processed two consecutive feature vectors and generated a single prediction for the following BOD observation. The LSTM layer maintained internal cell and hidden states and used input, forget, and output gates to regulate the retention and updating of temporal information. Its 32 hidden units, together with the hyperbolic-tangent activation, enabled the model to learn nonlinear relationships between the selected multifractal descriptors and BOD levels. The resulting predictive performance is depicted in **Fig. 4**, where the orange forecast line is plotted against the observed data.

To evaluate the predictive efficiency of the hybrid MFA-LSTM model against traditional Auto Regressive Integrated Moving Average (ARIMA) and Auto Regressive Fractionally Integrated Moving Average (ARFIMA) benchmarks, we utilized independent test datasets comprising 10-point (short-range) and 100-point (long-range) horizons. These data points were strictly excluded from the training phase to ensure an unbiased assessment of model generalization. The comparative performance was quantified using the RMSE metric,

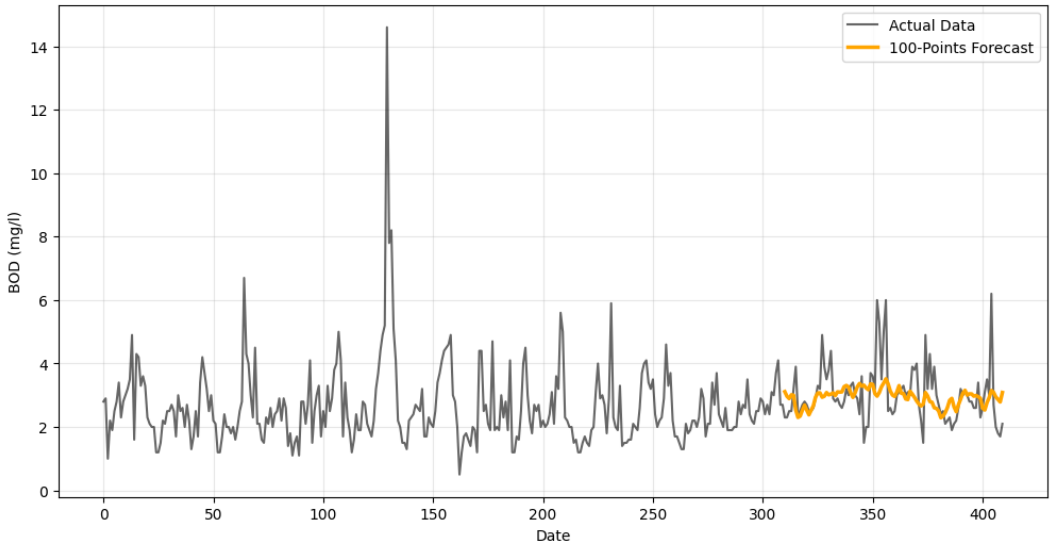


Fig. 4. LSTM forecasting (orange line) for the next 100 observations of BOD time series

allowing for a rigorous evaluation of each model's accuracy across different forecasting scales. The results of this analysis is presented in **Table 1**.

Table 1. **Evaluation of the prediction accuracy using RMSE metric**

Model	10 test data	100 test data
ARFIMA	1.2786	0.9875
ARIMA	1.2782	0.9878
MFA-LSTM	1.1808	0.8617

Based on the comparative RMSE results, the hybrid MFA-LSTM model demonstrates the highest predictive efficiency among the evaluated architectures for both short-term and long-term forecasting horizons. However, this superior accuracy is associated with increased computational complexity, requiring significant time and CPU resources for high-dimensional feature engineering and extensive hyperparameter optimization. Computational efficiency was evaluated by measuring training times on an Intel(R) Core(TM) i7-8550U processor. While the optimization of the MFA-LSTM architecture necessitated a 13-minute execution window to evaluate the complete hyperparameter grid, the ARIMA and ARFIMA benchmarks were processed nearly immediately. These results underscore the substantial overhead associated with high-dimensional feature engineering and neural network tuning compared to traditional statistical modeling. Furthermore, given that different training runs can yield models with varying internal characteristics, the 'black-box' nature of the LSTM must be acknowledged, necessitating careful interpretation of the model's decision-making process.

CONCLUSION

This research presents a predictive framework for BOD dynamics that integrates LSTM neural networks with multifractal analysis.

The methodology encompasses an assessment of stationarity, long-memory, and multifractal properties using the MFDFA algorithm, followed by a custom fractional-exponential approximation of the resulting singularity spectrum. By combining these

multifractal measures with statistical features within a sliding window approach, a high-dimensional feature vector was constructed for the LSTM model. Feature selection was optimized using a Random Forest Regressor, while model hyperparameters were refined through grid-search cross-validation.

The hybrid MFA-LSTM framework demonstrated superior predictive performance compared to all tested models, consistently achieving the lowest RMSE for the Dnipro River station. Specifically, the model yielded an RMSE of 1.1808 for a 10-step-ahead forecast and improved to 0.8617 for the 100-step-ahead horizon. While this performance advantage required greater computational time and CPU resources due to the intensive hyperparameter optimization process, the results validate the model's capacity to capture the complex, multi-scale dynamics of organic water pollution. Consequently, this methodology provides a robust analytical tool for the early detection and proactive management of river pollution events.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [M.B.]; methodology, [M.B., Y.S.]; validation, [Y.S.]; writing – original draft preparation, [M.B.]; writing – review and editing [M.B.]; supervision, [Y.S.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Donat-P. Häder, Anastazia T. Banaszak, Virginia E. Villafañe, Maite A. Narvarte, Raúl A. González, E. Walter Helbling. (2020) Anthropogenic pollution of aquatic ecosystems: Emerging problems with global implications. *Science of The Total Environment*, *713*, 136586. https://doi.org/10.1016/j.scitotenv.2020.136586

[2] Hu, J., Han, G., & Zhang, Q. (2025). Impacts of Environmental Change and Human Activities on Aquatic Ecosystems. *Water*, *17*(11), 1669. https://doi.org/10.3390/w17111669

[3] Zakharova, S.O, Arkhipova, K.K. (2018) Problemy Zabrudnennia Vodnykh Resursiv. Ekolohichnyi Aspekt. *World Science*, *8*(36), Vol.2, 58-62. (In Ukrainian) https://doi.org/10.31435/rsglobal_ws/30082018/6068

[4] Oberdorff, T. (2022) Time for decisive actions to protect freshwater ecosystems from global changes. *Knowl. Manag. Aquat. Ecosyst*., *423*, 19. https://doi.org/10.1051/kmae/2022017

[5] Horoshkova L., Menshov, O. I., Korniichuk, Y. D., Horoshkov, S. V., & Maslov, D. V. (2025). Ecological assessment and forecasting of surface water conditions in the Dnipro river in Zaporizhzhia. *Man and Environment. Issues of Neoecology*, *43*, 47-62. https://doi.org/10.26565/1992-4224-2025-43-04

[6] Jouanneau, Sulivan & Recoules, Loïc & Durand, Marie & Boukabache, A & Picot, V & Primault, Y & Lakel, Abdel & Sengelin, M & Barillon, Bruno & Thouand, Gerald. (2013). Methods for assessing biochemical oxygen demand (BOD): A review. *Water research,* 49C, 62-82. https://doi.org/10.1016/j.watres.2013.10.066

[7] Pattnaik, B. S., Pattanayak, A. S., Udgata, S. K., & Panda, A. K. (2021). Machine learning based soft sensor model for BOD estimation using intelligence at edge. *Complex & Intelligent Systems*, *7*(2), 961-976. https://doi.org/10.1007/s40747-020-00259-9
[8] Ching, P. M., So, R. H., & Morck, T. (2021). Advances in soft sensors for wastewater treatment plants: A systematic review. *Journal of Water Process Engineering*, *44*, 102367. https://doi.org/10.1016/j.jwpe.2021.102367
[9] Liu, Y., Li, K., & Huang, D. (2021, December). Prediction of Biochemical Oxygen Demand Based on VIP-PSO-Elman Model in Wastewater Treatment. In *Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications* (pp. 182-187). https://doi.org/10.1145/3491396.3506502
[10] Liang, Y., Ding, F., Liu, L., Yin, F., Hao, M., Kang, T., Zhao, C., Wang, Z., & Jiang, D. (2025). Monitoring water quality parameters in urban rivers using multi-source data and machine learning approach. *Journal of Hydrology*, *648*, 132394. https://doi.org/10.1016/j.jhydrol.2024.132394
[11] Bordun, M., & Mokrytska, O. (2024). Development of software and algorithmic equipment for prediction of river water pollution using fractal analysis methods. *Comput. Des. Syst. Theory Pract*, *6*(1), 178-188. https://doi.org/10.23939/cds2024.01.178
[12] Bordun, M., & Salapak, O. (2025). Mathematical models for the analysis and forecasting of river water pollution using the multifractal method. *Comput. Des. Syst. Theory Pract.*, *7*(1), 310-318. https://doi.org/10.23939/cds2025.01.310
[13] Yan, X., Cheng, H., Chi, S., Liu, S., & Zhu, Z. (2025). Spatiotemporal modeling of dissolved oxygen in a semi-enclosed water body with a LSTM-GRU hybrid approach. *Processes*, *13*(10), 3325. https://doi.org/10.3390/pr13103325
[14] State surface water monitoring data (n.d.) *State Agency of Water Resources of Ukraine.* (In Ukrainian). https://data.gov.ua/dataset/surface-water-monitoring
[15] Safitri, D., Mustafid, Ispriyanti, D., & Sugito. (2019, May). Gold price modeling in Indonesia using ARFIMA method. In *Journal of Physics: Conference Series* (Vol. 1217, No. 1, p. 012087). IOP Publishing. https://doi.org/10.1088/1742-6596/1217/1/012087
[16] Zhang, X., Zhang, G., Qiu, L., Zhang, B., Sun, Y., Gui, Z., & Zhang, Q. (2019). A modified multifractal detrended fluctuation analysis (MFDFA) approach for multifractal analysis of precipitation in Dongting Lake basin, China. *Water*, *11*(5), 891. https://doi.org/10.3390/w11050891
[17] He, M., Wu, S., Huang, B., Kang, C., & Gui, F. (2022). Prediction of total nitrogen and phosphorus in surface water by deep learning methods based on multi-scale feature extraction. *Water*, *14*(10), 1643. https://doi.org/10.3390/w14101643
[18] Chen, F., Sha, Y., Ji, H., Peng, K., & Liang, X. (2025). Integrating multifractal features into machine learning for improved prediction. *Fractal and Fractional*, *9*(4), 205. https://doi.org/10.3390/fractalfract9040205

АНАЛІЗ ТА ПРОГНОЗУВАННЯ ДИНАМІКИ ЗАБРУДНЕННЯ РІЧКОВИХ ВОД ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ ДОВГОЇ КОРОТКОСТРОКОВОЇ ПАМ'ЯТІ ТА МУЛЬТИФРАКТАЛЬНОГО АНАЛІЗУ

Михайло Бордун[1], Ярослав Соколовський[2]
[1] Кафедра інформаційних систем, Національний університет ім. Івана Франка, вул. Університетська, 1, Львів, 79007, УКРАЇНА
[2] Кафедра систем автоматизованого проектування, Національний університет «Львівська політехніка», вул. Степана Бандери, 12, м. Львів, 79013, УКРАЇНА

АНОТАЦІЯ

Вступ. Органічне забруднення води характеризується складною, нелінійною, нестаціонарною та самоподібною динамікою, яку традиційні моделі прогнозування часто не відтворюють. Метою дослідження є розроблення автоматизованої гібридної системи, що поєднує мультифрактальний аналіз і нейронне навчання для підвищення точності прогнозування біохімічного споживання кисню (БСК) та раннього виявлення екологічних ризиків.

Матеріали та методи. Ретроспективний часовий ряд БСК із річкового пункту моніторингу обробляли за допомогою автоматизованого конвеєра підготовки даних і моделювання. Ковзні вікна використовували для виділення локальних ознак і врахування довготривалих кореляцій. Метод мультифрактального детрендованого флуктуаційного аналізу (МФДФА, англ. MFDFA) застосовували для оцінювання узагальнених показників Херста та спектрів сингулярності в кожному вікні. За допомогою дробово-експоненційної апроксимації неперервні спектри перетворювали на дискретні коефіцієнти зі збереженням їхніх структурних характеристик. Мультифрактальні дескриптори поєднували зі статистичними показниками. Мережу довготривалої короткочасної пам’яті (ДКЧП, англ. LSTM) навчали за допомогою перехресної перевірки часових рядів та пошуку гіперпараметрів. Результати порівнювали з еталонними показниками авторегресивного інтегрованого ковзного середнього (АРІКС, англ. ARIMA – AutoRegressive Integrated Moving Average) та авторегресивного дробово інтегрованого ковзного середнього (АРДІКС, англ. ARFIMA – AutoRegressive Fractionally Integrated Moving Average).

Результати. Гібридна система МФДФА-ДКЧП забезпечила точніші й стабільніші прогнози, ніж базові моделі, для коротких і тривалих горизонтів. Дробово-експоненційні дескриптори надали інформацію про багатомасштабні флуктуації, персистентність, ширину спектра та асиметрію, що дало змогу ДКЧП ефективніше описувати нелінійну динаміку забруднення. Оптимізація гіперпараметрів збалансувала часову пам’ять, складність моделі та узагальнювальну здатність, зменшивши середньоквадратичну похибку.

Висновки. Дослідження підтверджує ефективність автоматизованої системи прогнозування якості води. Її інноваційність полягає у перетворенні неперервних мультифрактальних спектрів на інформативні дискретні ознаки за допомогою дробово-експоненційної апроксимації. У поєднанні з ДКЧП цей підхід підвищує надійність прогнозів і забезпечує природоохоронні органи інструментом раннього попередження та підтримки рішень для проактивного управління якістю води.

Ключові слова: ДКЧП, БСК, мультифрактальний аналіз, МФДФА, АРДІКС, спектр сингулярності.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
04 September, 2026	15 September, 2026	21 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 109–120
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 004.852:004.942:621.73

SURROGATE MODELING OF ELASTIC-PLASTIC SHEET METAL BENDING: A COMPARATIVE STUDY OF SURROGATE METHODS AND A DIAGNOSTIC TEST OF PHYSICS REGULARIZATION

Bohdan Petrovskyi
Department of Applied Mathematics,
Ivan Franko National University of Lviv,
1 Universytetska Street, Lviv, 79007, UKRAINE

Petrovskyi, B. (2026). Surrogate Modeling of Elastic-Plastic Sheet Metal Bending: A Comparative Study of Surrogate Methods and a Diagnostic Test of Physics Regularization. *Electronics and Information Technologies*, 35, 109–120. https://doi.org/10.30970/eli.35.8

ABSTRACT

Background. Sheet metal bending is a key manufacturing operation in which springback — the partial elastic recovery of the workpiece after unloading — causes deviations from the target geometry and must be predicted accurately. Repeated finite element simulations are computationally expensive for wide parametric studies, motivating fast surrogate models. This work compares surrogate modeling approaches for elastic-plastic sheet bending and evaluates the diagnostic value of a physics consistency check for detecting physically implausible neural surrogates, independently of whether physics regularization improves raw predictive accuracy.

Methods. A dataset of 39 568 finite element simulations was generated in COMSOL Multiphysics 6.2 by varying sheet thickness, punch displacement, and Young's modulus of a 6111-T4 aluminum alloy blank. Five classical surrogate methods and three neural network models (a pure data-driven baseline, a Hollomon-regularized model with correct parameters, and a control model with intentionally incorrect parameters) were compared over 5 random data splits. A sensitivity sweep varied the control model's hardening exponent between the true and the incorrect value to test how sharply the diagnostic responds.

Results. Classical non-parametric methods slightly outperform the neural surrogates on the dense smooth parameter grid: the random forest achieves a coefficient of determination of $R^2 > 0.9999$, compared to $R^2 \approx 0.999$ for the neural models. Differences in raw accuracy among the three neural models are statistically detectable across seeds (paired t-test) but tiny in magnitude ($\Delta R^2 < 0.1$ percentage point). The physics consistency metric, in contrast, separates the models by orders of magnitude: $R^2 = 0.999$ for the correct model versus $R^2 \approx -2.9$ for the incorrect one. The sensitivity sweep reveals a sharp threshold rather than gradual degradation: consistency remains near-perfect while the hardening exponent error stays small, then collapses once the exponent deviates substantially from its true value.

Conclusion. Standard accuracy metrics are insufficient to detect physically inconsistent neural surrogates, and the underlying diagnostic signal is thresholded rather than gradual. The physics consistency metric provides an independent diagnostic criterion that complements standard accuracy measures.

Keywords: physics-regularized neural networks; surrogate modeling; sheet metal bending; springback; Hollomon hardening law; random forest.

© 2026 Bohdan Petrovskyi. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Sheet metal forming is one of the key technological operations in mechanical engineering, aerospace, and automotive manufacturing. A major difficulty in sheet bending is springback, i.e., the partial elastic recovery of the workpiece after the punch is retracted. Springback causes deviations between the desired and the obtained geometry and therefore requires accurate prediction during process design. The finite element method (FEM) is the standard numerical tool for such problems [1, 2], but repeated FEM simulations become computationally expensive when a wide range of technological or material parameters must be investigated.

1.1. Surrogate models for metal forming

Surrogate and data-driven models for metal forming have been studied extensively. Early work used response surface methods and kriging [5, 6]. Neural networks were applied to springback prediction in V-bending [7] and air bending [8]. More recent work has employed Gaussian process regression, support vector machines, and ensemble methods for forming process optimization [9]. Machine learning methods have also been applied to fracture mechanics [10]. A comparative study of surrogate methods for sheet forming was presented in [11]. More recently, a machine learning-driven stress integration method was developed for anisotropic plasticity in sheet metal forming [21], and comparative studies of ANN and PINN approaches for sheet metal forming continue to appear [23].

Physics-informed and physics-regularized machine learning methods [3, 4] provide a way to embed physical knowledge into data-driven models. PINNs applied to elastoplasticity [12] and neural operator methods for multiscale inelastic problems [13], as well as physics-informed modeling of finite-strain plasticity [22], represent one direction. An alternative uses soft physics-based regularization terms without requiring the full PDE residual [14, 15]. The Hollomon hardening law has been incorporated into neural networks for material parameter identification [16] and stress prediction under cyclic loading [17].

1.2. Contribution of the present work

The present paper makes two contributions, both evaluated with multi-seed statistical support. First, it introduces a fake-physics control experiment as a diagnostic tool: a model with incorrect Hollomon parameters is trained alongside the physically consistent model, showing that standard accuracy metrics (coefficient of determination, R^2, and mean absolute error, MAE) cannot reliably detect physically inconsistent models, while a physics consistency metric separates them by orders of magnitude; a sensitivity sweep further shows that this diagnostic signal is thresholded rather than gradual. Second, it provides a systematic comparison of neural surrogate models against classical non-parametric methods (RBF interpolation, random forest, polynomial regression) on a benchmark sheet bending problem, showing that classical methods remain more accurate on this smooth three-parameter problem — so the value of the proposed model lies in its diagnostic behavior rather than in a raw predictive advantage.

2. MATERIALS AND METHODS

2.1. Problem formulation

The problem is a 2D plane-strain cross-section of a sheet metal bending process. The input vector is defined as:

$$\mathbf{x} = (t, \delta, E)^T \in \mathbb{R}^3, \tag{1}$$

where:

- t is sheet thickness (mm);
- δ is maximum punch displacement (mm);

- E is Young's modulus (GPa).

The output vector is defined as:

$$\mathbf{y} = (\sigma_{max}, u_{max}, \theta)^T \in \mathbb{R}^3, \tag{2}$$

where:

- σ_{max} is maximum von Mises stress (Pa);
- u_{max} is maximum displacement (m);
- θ is final bending angle after springback (° – deg).

2.2. FEM model for data generation

FEM simulations were performed in COMSOL Multiphysics 6.2 (Solid Mechanics module) using the benchmark sheet metal forming configuration [20], automated via a MATLAB script through the COMSOL LiveLink interface. The geometry is a 2D cross-section exploiting bilateral symmetry: a deformable sheet (L = 60 mm, w = 30 mm), a rigid punch (R_1 = 23.5 mm), and a rigid die (R_2 = 25 mm, shoulder R_3 = 4 mm, width 50 mm).

The blank is a 6111-T4 aluminum alloy with E_0 = 70.5 GPa, ν = 0.34, ρ = 2700 kg/m³, σ_0 = 194 MPa. Young's modulus E is varied as an input. Plastic hardening follows the Hollomon power law:

$$\sigma_{eq} = K \cdot \varepsilon_{np}, \quad K > 0, \quad n > 0, \tag{3}$$

with K = 550.4 MPa and n = 0.223 for the reference alloy. Friction coefficient μ = 0.1342; contact enforced with the augmented Lagrangian method [19]. The contact non-penetration conditions are:

$$g_n \geq 0, \quad \sigma_n \leq 0, \quad g_n \sigma_n = 0; \quad |\tau| \leq \mu |\sigma_n|. \tag{4}$$

These equations govern only the FEM simulations and are not imposed as residuals in the neural network loss.

2.3. Dataset construction

The dataset was assembled from multiple simulation runs with parameter ranges: $t \in [0.45, 1.50]$ mm, $\delta \in [18.0, 28.5]$ mm, $E \in [55.5, 175.5]$ GPa. The FEM solver advances the punch displacement through a monotonic arc-length continuation parameter; a run is retained only if this parameter reaches at least 0.95 of its target value, confirming that the solver tracked the loading path to (near-)completion without losing convergence. Runs that stalled below this threshold, typically for extreme parameter combinations that induce severe contact/plasticity nonlinearity, were discarded as non-converged rather than kept as unreliable partial solutions. The retained dataset contains N = 39 568 samples, all with von Mises stress above σ_0 = 194 MPa. The dataset is divided into training / validation / test (80% / 10% / 10%: 31 654 / 3 957 / 3 957 samples). Variables are standardized using statistics computed only on the training subset.

The raw von Mises stress exported via COMSOL LiveLink for MATLAB was rescaled by a fixed multiplicative factor to express MaxVonMises in SI Pascals, consistent with the units used for the Hollomon hardening parameters (K = 550.4 MPa) throughout the pipeline. This scaling was applied uniformly to the entire dataset before any train/validation/test split and before computing any of the standardization statistics above; it therefore does not affect relative comparisons (R^2, MAE, physics consistency) between surrogate models, only the absolute stress scale reported.

The Young's modulus range intentionally extends well beyond the physically realistic range for 6111-T4 aluminum (E_0 = 70.5 GPa) and partly overlaps typical values for steel. This was a deliberate choice to probe the surrogate's response surface over a wide input range, rather than to represent a physically realistic material variation; the corresponding predictions should be read as a numerical sensitivity study of the input–output map, not as material-specific forecasts outside the aluminum alloy's true elastic range.

2.4. Surrogate methods

Classical baselines. Five classical methods are evaluated:

1. Linear regression (multi-output OLS).
2. Polynomial regression of degree 2 (scikit-learn Pipeline).
3. Polynomial regression of degree 3 (scikit-learn Pipeline).
4. RBF interpolation — thin-plate spline (scipy RBFInterpolator, smoothing 10^{-3}), analogous to kriging.
5. Random forest — 300 regression trees (scikit-learn RandomForestRegressor).

Neural network baseline (Pure ML). Two hidden layers of 64 neurons, Swish activation, dropout 0.1, L^2 regularization 10^{-4}. Training: Adam (lr = 5×10^{-4}), batch 64, early stopping (patience 15), max 120 epochs.

Soft Hollomon-regularized model. Augments the network with a latent output $\hat{\varepsilon}_p$ (softplus-constrained). The total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \lambda_{\text{phys}}\mathcal{L}_{\text{phys}} + \alpha\mathcal{L}_{\text{bound}} + \beta\|\theta\|^2, \tag{5}$$

where $\mathcal{L}_{\text{phys}}$ penalizes the discrepancy between $\hat{\sigma}_{max}$ and Hollomon stress $\sigma_H = K(\hat{\varepsilon}_p + \varepsilon)^n$, normalized by s_σ. Penalty $\mathcal{L}_{\text{bound}}$ keeps $\hat{\varepsilon}_p \in [\varepsilon_{min}, \varepsilon_{max}]=[10^{-5}, 0.05]$. λ_{phys} annealed from 0 to 0.5; α = 10, β = 10^{-4}. Training: 25 pretraining + 80 physics-regularized epochs. Stress rescaled for numerical stability; $\hat{\varepsilon}_p$ is a numerically effective parameter, not the physical plastic strain.

The latent output $\hat{\varepsilon}_p$ is produced by a separate one-unit dense head with a softplus activation; this head shares the same two-layer trunk and the same input vector $\mathbf{x}$ as the three main output heads and receives no additional information.

The physics consistency metrics reported in Section 3.2 are defined as:

$$Physics\ MAE = \frac{1}{N}\sum_i |\sigma_{H,i} - \hat{\sigma}_{max,i}|, \tag{6}$$

$$Physics\ R^2 = 1 - \sum_i \left(\sigma_{H,i} - \hat{\sigma}_{max,i}\right)^2 \Big/ \sum_i \left(\sigma_{H,i} - \bar{\sigma}_H\right)^2, \tag{7}$$

$$Rel.error = \frac{1}{N}\sum_i |\hat{\sigma}_{max,i} - \hat{\sigma}_{H,i}|/(\sigma_{H,i} + \varepsilon), \tag{8}$$

where $\bar{\sigma}_H$ is the mean of σ_H over the test set and $\varepsilon = 10^{-6}$ avoids division by zero. All three compare the network's predicted stress $\hat{\sigma}_{max}$ against the Hollomon stress $\sigma_H = K(\hat{\varepsilon}_p + \varepsilon)^n$ implied by its own latent strain, over the N test-set samples.

Fake Hollomon control. Same as above but $n_{\text{fake}} = 0.8$ instead of $n = 0.223$ (K retained). Tests whether an inconsistent physical constraint degrades predictive accuracy.

Implementation. TensorFlow 2.x, Google Colaboratory (GPU); scikit-learn 1.8, scipy 1.17. Unless noted otherwise, the values shown are the mean over 5 independent random data splits (seeds 42, 123, 2024, 7, 2025); the training-fraction sensitivity (**Table 4**) and the fake-physics sensitivity sweep (Section 3.3) average over 3 of these seeds (42, 123, 2024), and all other single-model results use seed 42.

3. RESULTS AND DISCUSSION

3.1. Comparison of all surrogate methods

Table 1 reports R^2 values and bending angle MAE for all methods, averaged over 5 seeds ($N = 3\,957$ per split). **Table 2** shows MAE for all three outputs. RBF and random forest substantially outperform the neural models. The random forest attains $R^2 > 0.9999$ with angle MAE = 0.001°, compared to 0.273–0.289° for the neural models. The RBF interpolant achieves $R^2 = 0.9999$, angle MAE = 0.073°. Neural models match or slightly exceed polynomial regression of degree 2 and 3, confirming that the neural surrogate is a reasonable choice in higher-dimensional settings but does not outperform well-chosen non-parametric baselines on this three-parameter smooth problem.

Table 1. **Test R^2 and bending angle MAE ($N = 3\,957$, 10% split)**

Method	σ_{max} R^2	u_{max} R^2	θ R^2	θ MAE (°)
Linear regression	0.9982	0.9884	0.9978	0.598
Polynomial (deg. 2)	0.9992	0.9990	0.9994	0.301
Polynomial (deg. 3)	0.9993	0.9994	0.9997	0.222
RBF (thin-plate spline)	1.0000	0.9999	0.9999	0.073
Random forest (300 trees)	1.0000	1.0000	1.0000	0.001
Neural net (Pure ML)	0.9991	0.9992	0.9995	0.281
+ Hollomon (true K, n)	0.9990	0.9993	0.9995	0.273
+ Hollomon (fake $n = 0.8$)	0.9987	0.9992	0.9995	0.289

Table 2. **Mean absolute errors (MAE) for all surrogate methods on the test set**

Method	σ_{max} MAE (Pa)	u_{max} MAE (m)	θ MAE (°)
Linear regression	1.44×10^{6}	2.50×10^{-4}	0.598
Polynomial (deg. 2)	6.42×10^{5}	6.88×10^{-5}	0.301
Polynomial (deg. 3)	6.16×10^{5}	5.26×10^{-5}	0.222
RBF (thin-plate spline)	4.34×10^{4}	1.49×10^{-5}	0.073
Random forest (300 trees)	4.25×10^{3}	2.39×10^{-7}	0.001
Neural net (Pure ML)	7.59×10^{5}	6.45×10^{-5}	0.281
+ Hollomon (true K, n)	8.61×10^{5}	5.95×10^{-5}	0.273
+ Hollomon (fake $n = 0.8$)	1.00×10^{6}	6.52×10^{-5}	0.289

3.2. Physical consistency

Although all three neural models achieve comparable predictive accuracy, they differ substantially in physical consistency. **Table 3** compares the predicted stress $\hat{\sigma}_{\max}$ with the Hollomon stress σ_H computed from the latent output. To test how sharply this diagnostic responds to the degree of physical inconsistency, the control model was retrained with the hardening exponent n swept between the true value (0.223) and the fake value (0.8), on 3 seeds per value.

Table 3. **Physical consistency vs. hardening exponent *n* used in the control model (*mean* ± *std* over 3 seeds: 42, 123, 2024). True value *n* = 0.223; fake control *n* = 0.8**

Hollomon exponent n	*Physics* MAE (Pa)	*Physics* R^2	*Rel. error*
n = 0.223 (true)	1.43×10⁶ ± 5.4×10⁵	0.999 ± 0.001	1.0% ± 0.4%
n = 0.35	1.07×10⁶ ± 6.4×10⁵	0.999 ± 0.002	0.7% ± 0.4%
n = 0.5	8.26×10⁷ ± 4.3×10⁷	−5.25 ± 4.66	35.2% ± 8.5%
n = 0.65	4.34×10⁷ ± 3.4×10⁶	−1.48 ± 0.40	33.1% ± 3.5%
n = 0.8 (fake)	5.51×10⁷ ± 3.8×10⁵	−2.90 ± 0.65	67.0% ± 4.1%

Averaged over 3 seeds, the model with the correct exponent achieves physics R^2 = 0.999±0.001 and relative error 1.0%±0.4%, while the fake-physics control (n = 0.8) achieves *Physics* R^2 = −2.90±0.65 and *Relative error* 67.0%±4.1% (**Table 3**). The sensitivity sweep reveals a sharp threshold rather than a gradual decline: consistency remains near-perfect at n = 0.35 (*Physics* R^2 = 0.999±0.002), a 57% *Relative error* in the exponent itself, but collapses once n reaches 0.5 (*Physics* R^2 = −5.25±4.66) and stays poor thereafter. The latent strain for the true model is consistent with the FEM benchmark observation of ≈2% maximum equivalent plastic strain [20]; for $n \geq 0.5$ the latent strain substantially exceeds the soft upper bound $\varepsilon_{max} = 0.05$. **Fig. 1** and **Fig. 2** illustrate the scatter and loss histories for the true and fake endpoints.

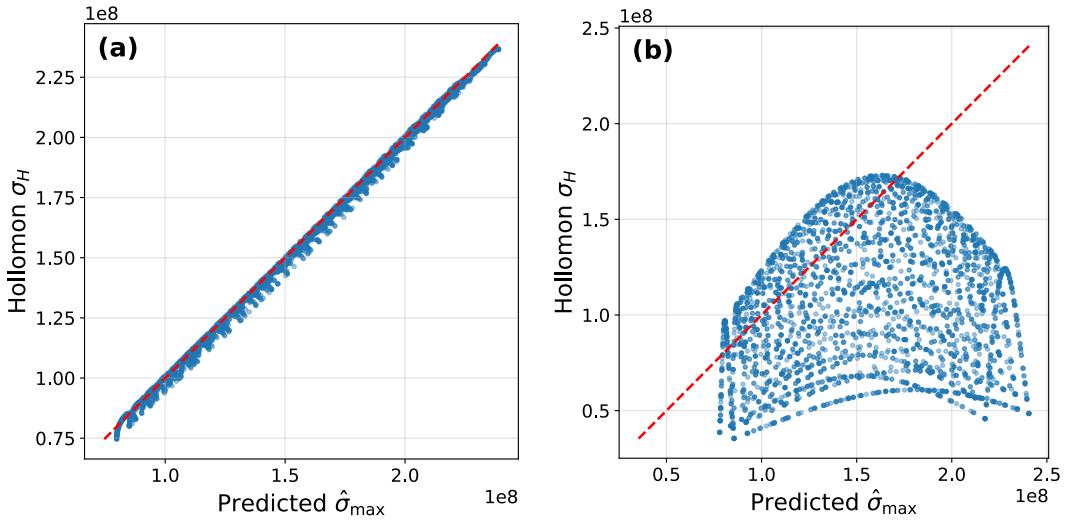


Fig. 1. Predicted stress $\hat{\sigma}_{max}$ vs. Hollomon stress σ_H: (a) true K, n; (b) fake $n = 0.8$. Dashed line — perfect agreement

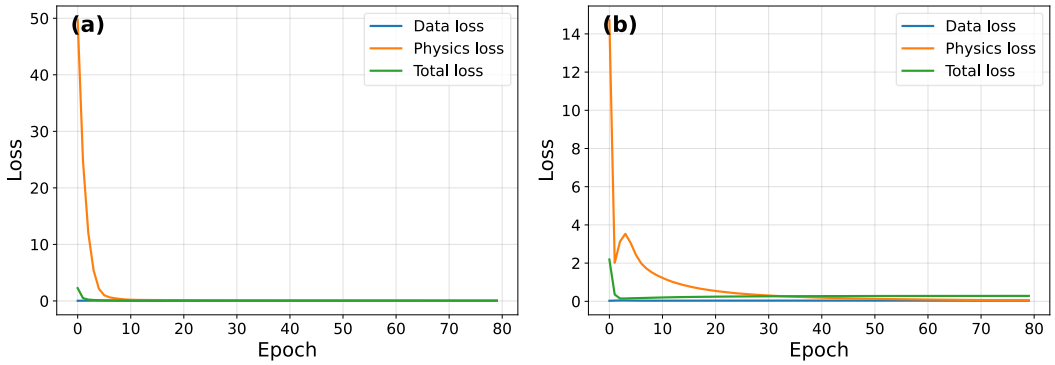


Fig. 2. Training loss histories: (a) true model; (b) fake model. Physics loss (orange) plateaus higher for the fake model

3.3. Accuracy does not imply physical consistency

All three neural models achieve $R^2 \approx 0.999$ regardless of whether the Hollomon parameters are correct, incorrect, or absent. Differences in raw accuracy across the three models are statistically detectable over 5 seeds (paired t-test on R^2 of σ_{max}: Pure ML vs. true-Hollomon, $p = 0.22$, not significant; Pure ML vs. fake-Hollomon, $p = 0.014$; true- vs. fake-Hollomon, $p = 0.016$) but are tiny in absolute magnitude ($\Delta R^2 < 0.1$ percentage point) and practically negligible for model selection based on accuracy alone. This illustrates a general risk in data-driven modeling: high predictive accuracy does not guarantee physical consistency [4]. The physics consistency metric (**Table 3**), in contrast, separates the models by orders of magnitude and remains diagnostic across a wide range of hardening-exponent perturbations.

3.4. Sensitivity to training set size

Table 4 shows mean R^2 over 3 seeds for training fractions from 5% to 100%. Standard deviation $\sigma < 0.001$ in all entries; maximum $\sigma = 0.0007$ (u_{max}, Hollomon, 5%). **Fig. 3** illustrates the results.

Both models maintain $R^2 \approx 0.999$ across all fractions. The pure neural network is marginally more stable at 5–10%, while the Hollomon model catches up from 25% onward. Physics regularization does not provide a clear advantage in data efficiency.

Table 4. **Mean test R^2 over 3 seeds vs. training fraction. Max $\sigma = 0.0007$ (u_{max}, Hollomon, 5%); all other entries: $\sigma < 0.0005$**

Output	Model	5%	10%	25%	50%	100%
σ_{max}	Pure ML	0.9989	0.9989	0.9989	0.9990	0.9989
	Hollomon	0.9983	0.9983	0.9988	0.9989	0.9989
u_{max}	Pure ML	0.9989	0.9991	0.9992	0.9993	0.9992
	Hollomon	0.9974	0.9985	0.9991	0.9993	0.9993
θ	Pure ML	0.9993	0.9995	0.9995	0.9994	0.9995
	Hollomon	0.9989	0.9991	0.9995	0.9995	0.9995

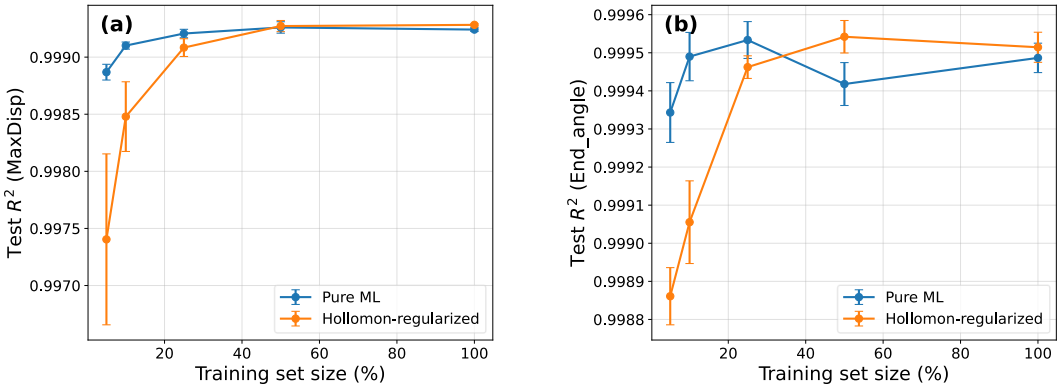


Fig. 3. Test R² vs. training set size (*mean* ± *std*, 3 seeds): (a) u_{max}; (b) θ

3.5. When classical baselines are competitive

RBF and random forest outperform the neural surrogate on this problem [11, 18]. This is consistent with known behavior for smooth low-dimensional response surfaces on regular grids. The neural surrogate would offer advantages for higher input dimensionality (e.g., including Poisson's ratio, friction coefficient, tool geometry), irregular sampling, or when explicit physical constraints on outputs are required.

3.6. Interpretation of the Hollomon regularization

The Hollomon regularization links the predicted maximum von Mises stress to a latent scalar strain through a global power law. In a real bending problem, the stress distribution is nonuniform and the maximum stress is not uniquely determined by the scalar Hollomon law at a single representative strain. The latent strain $\hat{\varepsilon}_p$ is therefore an effective model parameter, not a physically measurable quantity, and is not validated against FEM plastic strain fields. Nevertheless, the fake-physics experiment confirms that the physics consistency check is informative.

3.7. Limitations

The following limitations apply:

1. Dataset generated numerically; accuracy bounded by FEM fidelity.
2. Latent plastic strain not validated against FEM fields.
3. Only global scalar quantities predicted; no field reconstruction.
4. Neural surrogate does not outperform classical baselines on this smooth three-parameter problem.
5. Dataset grid is non-uniform across simulation runs.
6. Extrapolation outside the parameter range is not assessed.

4. CONCLUSIONS

A systematic comparison of surrogate modeling methods for elastic-plastic sheet metal bending has been presented. Five classical methods and three neural network models were evaluated on a dataset of 39 568 FEM simulations. The main findings are:

1. Classical baselines outperform neural surrogates on the dense smooth parameter grid. The random forest attains $R^2 > 0.9999$, the RBF interpolant $R^2 = 0.9999$, compared to $R^2 \approx 0.999$ for the neural models.
2. Physics regularization does not improve data efficiency: both pure and Hollomon-regularized models maintain comparable accuracy across training set fractions from 5% to 100%.

3. Standard accuracy metrics cannot detect physically inconsistent models. Over 3 seeds, the fake-physics control exhibits 67.0% ± 4.1% physics consistency error and physically unrealistic latent strain values, while the correct model achieves 1.0% ± 0.4% physics error. Differences in raw predictive accuracy among the three neural models, while statistically detectable across seeds, are tiny in absolute magnitude ($\Delta R^2 < 0.1$ percentage point); the physics consistency metric, in contrast, separates them by orders of magnitude.
4. The diagnostic signal is thresholded, not gradual. A sensitivity sweep over the control model's hardening exponent shows that physics consistency remains near-perfect while the exponent error stays below ~57% of its true value, then collapses sharply once the error grows further — a practical implication for how large a parameter mismatch this diagnostic can be expected to catch.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author expresses sincere gratitude to Andrii Pereimybida, Associate Professor at the Department of Applied Mathematics, Ivan Franko National University of Lviv, for helpful consultations, discussion of the manuscript structure, and assistance with interpreting the results.

The author received no financial support for the research, writing, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The author declares that the research was conducted in the absence of any conflict of interest.

CODE AND DATA AVAILABILITY.

The Python training/analysis code is available at https://colab.research.google.com/drive/1ys2mN5-h_uRl6N6BzixdI7aaAY8sEjT7?usp=sharing. The COMSOL model, MATLAB data-generation script, and raw simulation dataset are available from the author upon reasonable request.

Author has read and agreed to the published version of the manuscript.

REFERENCES

[1] Belytschko, T., Liu, W. K., Moran, B., & Elkhodary, K. (2014). *Nonlinear finite elements for continua and structures* (2nd ed.). Wiley. https://www.wiley.com/en-us/Nonlinear+Finite+Elements+for+Continua+and+Structures,+2nd+Edition-p-9781118632703

[2] Zienkiewicz, O. C., & Taylor, R. L. (2005). *The finite element method* (5th ed.). Butterworth-Heinemann. https://doi.org/10.1016/B978-075066431-8.50196-X

[3] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[4] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

[5] Naceur, H., Ben-Elechi, S., Batoz, J. L., & Knopf-Lenoir, C. (2008). Response surface methodology for the rapid design of aluminum sheet metal forming parameters. *Materials & Design*, *29*(4), 781–790. https://doi.org/10.1016/j.matdes.2007.01.018
[6] Breitkopf, P., Naceur, H., Rassineux, A., & Villon, P. (2005). Moving least squares response surface approximation: Formulation and metal forming applications. *Computers & Structures*, *83*(17–18), 1411–1428. https://doi.org/10.1016/j.compstruc.2004.07.011
[7] Kazan, R., Firat, M., & Tiryaki, A. E. (2009). Prediction of springback in wipe-bending process of sheet metal using neural network. *Materials & Design*, *30*(2), 418–423. https://doi.org/10.1016/j.matdes.2008.05.033
[8] Inamdar, M. V., Date, P. P., & Desai, U. B. (2000). Studies on the prediction of springback in air vee bending of metallic sheets using an artificial neural network. *Journal of Materials Processing Technology*, *108*(1), 45–54. https://doi.org/10.1016/S0924-0136(00)00588-4
[9] Spathopoulos, S. C., & Stavroulakis, G. (2020). Springback prediction in sheet metal forming using FEA and ANN. *Applied Mechanics*, *1*(2), 97–110. https://doi.org/10.3390/applmech1020007
[10] Liu, X., Athanasiou, C. E., Padture, N. P., Sheldon, B. W., & Gao, H. (2020). A machine learning approach to fracture mechanics problems. *Acta Materialia*, *190*, 105–112. https://doi.org/10.1016/j.actamat.2020.03.016
[11] Hamdaoui, M., Le Quilliec, G., Breitkopf, P., & Villon, P. (2014). POD surrogates for real-time multi-parametric sheet metal forming problems. *International Journal of Material Forming*, *7*(3), 337–358. https://doi.org/10.1007/s12289-013-1132-0
[12] Haghighat, E., Raissi, M., Moure, A., Gomez, H., & Juanes, R. (2021). A physics-informed deep learning framework for inversion and surrogate modeling. *Computer Methods in Applied Mechanics and Engineering*, *379*, 113741. https://doi.org/10.1016/j.cma.2021.113741
[13] Yin, M., Zhang, E., Yu, Y., & Karniadakis, G. E. (2022). Interfacing finite elements with deep neural operators for fast multiscale modeling. *Computer Methods in Applied Mechanics and Engineering*, *402*, 115027. https://doi.org/10.1016/j.cma.2022.115027
[14] Zhang, D., Lu, L., Guo, L., & Karniadakis, G. E. (2019). Quantifying total uncertainty in physics-informed neural networks. *Journal of Computational Physics*, *397*, 108850. https://doi.org/10.1016/j.jcp.2019.07.048
[15] Tandale, S. B., Bamer, F., Markert, B., & Stoffel, M. (2022). Physics-based self-learning recurrent neural network for viscoplastic FEM response. *Computer Methods in Applied Mechanics and Engineering*, *401*, 115668. https://doi.org/10.1016/j.cma.2022.115668
[16] Versino, D., Tonda, A., & Bronkhorst, C. A. (2017). Data driven modeling of plastic deformation. *Computer Methods in Applied Mechanics and Engineering*, *318*, 981–1004. https://doi.org/10.1016/j.cma.2017.02.016
[17] Gorji, M. B., Mozaffar, M., Heidenreich, J. N., Cao, J., & Mohr, D. (2020). On the potential of recurrent neural networks for modeling path dependent plasticity. *Journal of the Mechanics and Physics of Solids*, *143*, 103972. https://doi.org/10.1016/j.jmps.2020.103972
[18] Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate* modelling*: A practical guide*. Wiley. https://doi.org/10.1002/9780470770801

[19] Dyyak, I. I., & Prokopyshyn, I. I. (2010). Domain decomposition schemes for frictionless multibody contact problems of elasticity. In *Numerical Mathematics and Advanced Applications 2009* (pp. 301–308). Springer. https://doi.org/10.1007/978-3-642-11795-4_31
[20] COMSOL AB. (2023). Sheet Metal Forming. In *COMSOL Multiphysics Application Library, Nonlinear Structural Materials Module* (v6.2). https://www.comsol.com/model/sheet-metal-forming-11208
[21] Fazily, P., & Yoon, J. W. (2023). Machine learning-driven stress integration method for anisotropic plasticity in sheet metal forming. *International Journal of Plasticity*, *166*, 103642. https://doi.org/10.1016/j.ijplas.2023.103642
[22] Niu, S., Zhang, E., Bazilevs, Y., & Srivastava, V. (2023). Modeling finite-strain plasticity using physics-informed neural network and assessment of the network performance. *Journal of the Mechanics and Physics of Solids*, *172*, 105177. https://doi.org/10.1016/j.jmps.2022.105177
[23] Munzone, F., Hazrati, J., Hakvoort, W., & van den Boogaard, T. (2024). Comparative study of artificial neural network and physics-informed neural network application in sheet metal forming. *Materials Research Proceedings*, *41*, 2278–2288. https://doi.org/10.21741/9781644903131-251

АПРОКСИМАЦІЙНЕ МОДЕЛЮВАННЯ ПРУЖНО-ПЛАСТИЧНОГО ЗГИНАННЯ ЛИСТОВОГО МЕТАЛУ: ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ АПРОКСИМАЦІЙНИХ МЕТОДІВ ІЗ ДІАГНОСТИЧНОЮ ПЕРЕВІРКОЮ ФІЗИЧНОЇ РЕГУЛЯРИЗАЦІЇ

Богдан Петровський
Кафедра прикладної математики,
Львівський національний університет імені Івана Франка,
вул. Університетська, 1, Львів, 79007, УКРАЇНА

РЕЗЮМЕ

Вступ. Згинання листового металу є ключовою технологічною операцією, після якої відбувається пружне відновлення форми (англ. springback) — часткове пружне повернення заготовки після зняття навантаження, яке спричиняє відхилення від цільової геометрії деталі. Багаторазові симуляції методом скінченних елементів (МСЕ; англ. FEM) є обчислювально затратними для широких параметричних досліджень. Роботу присвячено порівнянню підходів до апроксимаційного моделювання та оцінці діагностичної цінності перевірки фізичної узгодженості для виявлення фізично неправдоподібних моделей незалежно від того, чи покращує регуляризація саму точність.

Матеріали та методи. Набір даних із 39 568 симуляцій МСЕ сформовано в COMSOL Multiphysics 6.2 шляхом варіювання товщини листа, переміщення пуансона та модуля Юнга заготовки зі сплаву 6111-T4. П'ять класичних методів та три моделі нейронних мереж (базова модель, керована лише даними; модель з регуляризацією за законом Холломона з коректними параметрами; контрольна модель із навмисно неправильними параметрами) порівняно на 5 випадкових розбиттях даних. Додатково просканована чутливість діагностики до величини «неправильності» показника зміцнення в контрольній моделі — від істинного значення до хибного.

Результати. Класичні непараметричні методи перевершують нейронну апроксимаційну модель на щільній гладкій параметричній сітці: випадковий ліс досягає

$R^2 > 0{,}9999$ проти $R^2 \approx 0{,}999$ для нейронних моделей. Різниці в точності між трьома нейронними моделями статистично виявляються між seeds (парний *t*-тест), але дуже малі за величиною ($\Delta R^2 < 0{,}1$ відсоткового пункту). Натомість метрика фізичної узгодженості розділяє моделі на порядки: $R^2 = 0{,}999$ для коректної моделі проти $R^2 \approx -2{,}9$ для некоректної. Сканування чутливості показує різкий пороговий ефект, а не поступову деградацію: узгодженість залишається майже ідеальною, доки похибка показника зміцнення невелика, і різко руйнується після суттєвого відхилення.

Висновки. Стандартні показники точності є недостатніми для виявлення фізично неузгоджених нейронних моделей, а сам діагностичний сигнал має пороговий, а не поступовий характер. Метрика фізичної узгодженості є незалежним діагностичним критерієм, який доповнює стандартні показники точності.

Ключові слова: фізично-регуляризовані нейронні мережі; апроксимаційне моделювання; згинання листового металу; пружне відновлення форми; закон зміцнення Холломона; ансамблевий метод випадкових дерев рішень.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
17 July, 2026	30 August, 2026	03 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 121–130
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.93:621.3.049.77

DISTRIBUTED ANOMALY DETECTION SYSTEM IN MANUFACTURING USING EDGE AI AND COMPUTER VISION

***Olha Shymchyshyn*[1]*, Kostiantyn Pidhornyi*[1]*, Maryan Shymchyshyn*[2]
[1] *Lviv Polytechnic National University,*
12 Stepana Bandery Street, Lviv, 79013, UKRAINE
[2] *Lviv Branch of Private Higher Education Establishment "European University",*
5 Kushevycha Street, Lviv, 79019, UKRAINE

Shymchyshyn, O., Pidhornyi, K. & Shymchyshyn, M. (2026). Distributed Anomaly Detection System in Manufacturing Using Edge AI and Computer Vision. *Electronics and Information Technologies*, 35, 121–130. https://doi.org/10.30970/eli.35.9

ABSTRACT

Background. Modern automated production facilities are characterized by a high level of robotization and the drive to minimize human involvement in routine operations. At the same time, the complete elimination of the operator is unacceptable due to safety requirements and the need for decision-making in the event of anomalies. The use of centralized computer vision systems results in significant network load, limiting their scalability. One promising direction for solving this problem is the application of Edge AI technology.

Materials and Methods. A three-tier architecture for an anomaly detection system is proposed. The first tier is implemented based on the ESP32-P4 edge nodes with digital cameras, which perform local object detection using quantized computer vision models. Only structured metadata is transmitted to the central cluster via the CAN bus. The second tier performs motion trajectory analysis based on root mean square (RMS) deviation. The third tier implements an operator notification subsystem with automatic generation of an event video replay.

Results and Discussion. Offloading object detection to edge nodes significantly reduces network load and ensures system scalability regardless of the number of cameras. The use of the ESP32-P4 enables real-time video processing at up to 30 frames/s with a power consumption of less than 500 mW per node. RMS analysis ensures the detection of deviations from the reference trajectory in less than 1 s. After an anomaly is recorded, the operator receives a push notification and a 5-second video clip for rapid situation assessment.

Conclusion. The proposed architecture demonstrates the feasibility of practical Edge AI utilization for building scalable production process monitoring systems. The combination of local computer vision, transmission of structured data via the CAN bus, and centralized statistical analysis ensures continuous technological process control, reduces network load, and shortens response time to anomalous events. Further research will focus on the adaptive tuning of RMS thresholds and the implementation of federated learning for the models.

Keywords: edge computing, computer vision, ESP32-P4, anomaly detection, CAN bus, industrial automation

INTRODUCTION

Modern automated manufacturing enterprises increasingly operate with a minimal staff presence on the production floor. Robotic manipulators, conveyors, and automated guided vehicles (AGVs) perform the majority of physical operations, while personnel take

© 2026 Olha Shymchyshyn et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

on supervisory functions. However, the complete elimination of a responsible person is considered unacceptable both from a safety perspective and for regulatory reasons [1, 2].

The human-computer interaction scientific literature emphasizes the need for reliable two-way collaboration frameworks that allow operators to identify system model biases and increase decision-making reliability [3]. Traditional closed-circuit television (CCTV) systems require constant human attention and scale poorly for enterprises with dozens or hundreds of cameras. Centrally processed computer vision (CV) systems running on a server create significant network load when processing high-framerate video streams from multiple cameras simultaneously. This contradiction between automation requirements and the need for human supervision creates the demand for a system that combines continuous machine monitoring while retaining the operator as the ultimate arbiter upon detecting deviations.

The Edge AI approach, which involves performing data processing using machine learning models directly on embedded devices close to the data source, significantly reduces network load and processing latency, thereby ensuring system scalability [4]. Provided that object detection is performed directly on peripheral camera nodes, only compact structured metadata — specifically object identifiers, bounding box coordinates, and timestamps — are transmitted to the central cluster. This approach substantially reduces the volume of transmitted data and lowers network bandwidth requirements. The implementation of this paradigm has been made possible by the emergence of microcontrollers with specialized hardware accelerators for neural network computations, notably the ESP32-P4 with a dual-core RISC-V architecture and an integrated Image Signal Processor (ISP) accelerator. The purpose of this work is to develop a distributed computer vision system architecture for real-time anomaly detection in production objects based on edge data processing (Edge AI) and centralized cluster analysis with operator notification generation.

MATERIALS AND METHODS

General System Architecture

The proposed system comprises three hierarchical levels, as illustrated in **Fig. 1**. The first level consists of distributed edge nodes deployed across the enterprise. The second level is a centralized data processing cluster. The third level represents the operator notification subsystem.

Level 1 — Edge Nodes

Each edge node is implemented as an intelligent computer vision module consisting of a digital camera and an ESP32-P4 microcontroller. The primary purpose of the node is local video stream processing, object detection, and transmitting only compact structured metadata to the central cluster instead of the raw video stream.

The choice of the ESP32-P4 platform is driven by its architectural features focused on edge data processing tasks. The microcontroller is built on two 32-bit RISC-V cores with a clock frequency of up to 400 MHz and features an integrated Image Signal Processor (ISP) module that performs hardware preprocessing of video data. Optimized ESP-NN libraries are used to accelerate neural network model execution, enabling efficient execution of quantized models using integer arithmetic (INT8). This approach ensures real-time local execution of computer vision models with low power consumption of the edge node [5].

Object detection is performed using pre-trained and quantized computer vision models deployed directly on the edge node. Model training is carried out on external computational resources using representative samples of a specific production process, after which the model is optimized and converted for execution on the microcontroller.

The choice of the neural network architecture is determined by the specifics of the production process, the required detection accuracy, and the available computational resources of the edge node. Since Edge AI operates under constrained RAM and power

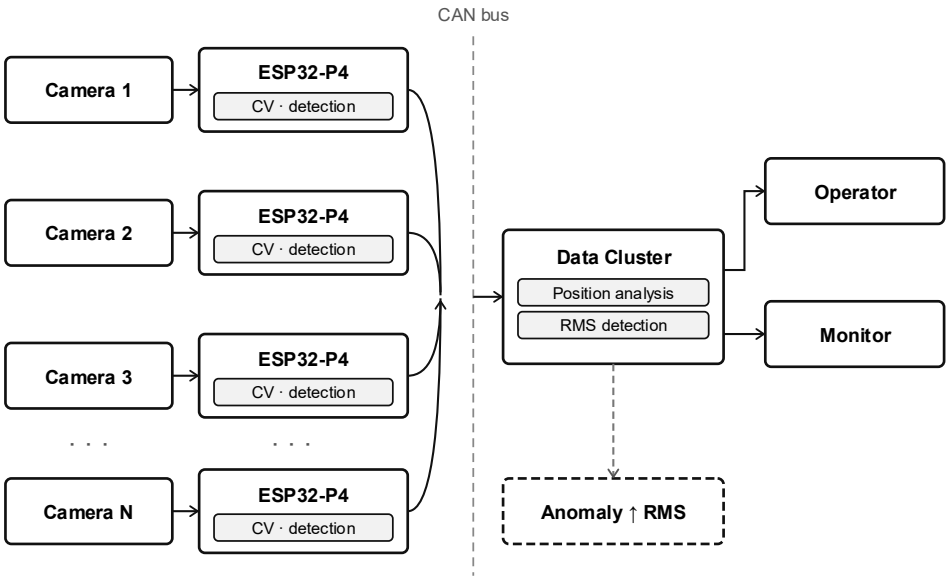


Fig. 1. Three-tier distributed architecture of the Edge AI-based anomaly detection system: (1) edge nodes (camera + ESP32-P4); (2) CAN bus; (3) data processing cluster; (4) operator notification system

consumption conditions, key criteria for model selection include performance, processing latency, object localization accuracy, and the amount of PSRAM used.

Comparative characteristics of the most common models suitable for implementation on the ESP32-P4 platform are presented in **Table 1**. The presented characteristics are generalized based on data [6–8] and may vary depending on the model configuration, input image resolution, and hardware platform parameters. As can be seen from **Table 1**, the use of more complex models ensures higher object localization accuracy, but is accompanied by an increase in processing latency and memory consumption. For high-speed production lines, the application of specialized regression models or NanoDet-m is appropriate, whereas for tasks involving complex scenes with a large number of objects, models of the YOLO family are more suitable. Therefore, the choice of a specific model is

Table 1. **Comparative characteristics of computer vision models for implementation on edge nodes**

Model	FPS	Latency, ms	Localization accuracy	PSRAM load	Recommended scenario
Custom Regressor	~45 – 60	< 20	Medium	Minimal	Fast lines, single stable object.
NanoDet-m	~18 – 25	40 – 55	High	Medium	Optimal for sorting parts.
SSD Lite (MobNetV2)	~12 – 18	60 – 80	High	High	Standard industrial tasks.
YOLOv5n	~9 – 14	70 – 110	Very high	High	Detection of multiple small objects.
YOLOv8n	~6 – 10	110 – 160	Very high	Very high	Complex scenes where accuracy is critical.
YOLO26n	~12 – 16	65 – 85	Extremely high	Very high	Precise coordinates + OBB.

determined by a trade-off between performance, object localization accuracy, processing latency, and the amount of available hardware resources of the edge node.

Inter-level Communication

To organize data exchange between edge nodes and the central processing cluster, the Controller Area Network (CAN) bus was chosen. This choice is driven by a number of characteristics critical for the system's operation in an industrial environment. Specifically, CAN ensures deterministic message delivery with a bounded maximum latency, which is essential for real-time systems. In addition, the protocol is characterized by high immunity to electromagnetic interference, the presence of a built-in multi-master arbitration mechanism without the need for network switches, and the widespread availability of hardware implementations and industrial transceivers [9].

The system utilizes the CAN 2.0B standard with an extended 29-bit identifier, allowing for flexible addressing of nodes and message types. Data transmission is carried out in frames of up to 8 bytes in length, the structure of which is optimized for transmitting compact metadata obtained at the edge level. The frame data field transmits: node identifier (1 byte), object class (1 byte), normalized X and Y centroid coordinates (2 bytes each, Q8.8 fixed-point format), and a relative timestamp (2 bytes). Representing coordinates in the Q8.8 fixed-point format provides a trade-off between positioning accuracy and message compactness, which is important for CAN networks with a small frame size. At a detection frequency of 20 frames/s, a single node generates about 160 bytes/s of overhead data, whereas transmitting an uncompressed VGA video stream requires over 18 MB/s. Thus, using local detection reduces network traffic by more than a factor of 10^5 (the order of magnitude depends on resolution, frame rate, and compression parameters).

Given that the CAN bus operates as a broadcast medium without built-in security protocols, addressing network security and message integrity is paramount, particularly when transmitting safety-critical data. In industrial environments, unprotected CAN frames are vulnerable to message spoofing, frame injection, and replay attacks. To mitigate these threats, the system architecture incorporates node authentication mechanisms, such as Secure On-Board Communication (SecOC) or Hash-based Message Authentication Codes (HMAC), appended to critical metadata payloads. This is especially vital when the CAN bus is leveraged for functional safety functions, such as transmitting emergency stop (E-stop) signals or monitoring hazardous operator zones. Ensuring data integrity and authenticating edge node identities guarantees that emergency overrides cannot be maliciously intercepted or altered, complying with industrial functional safety standards.

The proposed message structure allows transmitting only the information necessary for further analysis without forwarding the video stream. This significantly reduces the network load and ensures system scalability as the number of edge nodes increases. Unlike transmitting the full video stream, the proposed approach involves transmitting exclusively structured results of local processing, which reduces the volume of network traffic by several orders of magnitude. As a result, the performance of the central cluster is practically independent of the number of cameras, since the server analyzes only compact object descriptions rather than primary images.

Level 2 — Data Processing Cluster and Anomaly Detection

For each tracked object, the reference trajectory is formed during the system calibration stage under normal operating conditions. During calibration, the object's coordinates are recorded over several cycles of normal production line operation. The reference trajectory is generated by averaging the corresponding trajectories obtained during these cycles and is used as a baseline model of normal object motion. The use of a baseline model representing normal process behavior is consistent with the general principles of statistical process monitoring [10].

Anomaly detection is carried out by comparing the current coordinates of the object with the corresponding reference values. For each frame *i* within a sliding time window of duration τ, the instantaneous deviation $d(t_i)$ is computed as the Euclidean distance between the actual coordinates (x_{actual}, y_{actual}) and the planned reference coordinates $(x_{planned}, y_{planned})$:

$$d(t_i) = \sqrt{\left[\left(x_{planned} - x_{actual}\right)^2 + \left(y_{planned} - y_{actual}\right)^2\right]}.$$

The root mean square (RMS) deviation $RMS(t)$ is evaluated over a sample of $N = \tau \cdot f$, measurements (where f is the detection frame rate) within the sliding window τ:

$$RMS(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[d(t_i)]^2}.$$

The selection of the anomaly detection threshold RMS_{thresh} is grounded in statistical process control principles [10]. During the system calibration phase under standard operating conditions, the baseline distribution of the sliding RMS value is evaluated to calculate its historical mean μ_{RMS} and standard deviation σ_{RMS}. The detection threshold is then defined using the k-sigma statistical rule:

$$RMS_{thresh} = \mu_{RMS} + k \cdot \sigma_{RMS},$$

where k is a tuning parameter (typically $k \in [3, 6]$) that determines the system's sensitivity trade-off. Increasing k lowers the False Positive Rate (FPR), preventing nuisance alerts caused by minor vibration or lighting changes, but increases the risk of False Negatives (FN). Conversely, a smaller k maximizes sensitivity to micro-deviations at the expense of potential false alarms. The duration of the sliding window τ (and corresponding sample size $N = \tau \cdot f$) acts as a low-pass filter: longer windows smooth out transient sensor noise, whereas shorter windows ensure sub-second anomaly response times.

Exceeding the set RMS threshold ($RMS(t) > RMS_{thresh}$) is interpreted as an abnormal state of the object.

Simultaneously with processing the positional data, the cluster maintains a video ring buffer that receives a parallel low-bitrate H.264 video stream. This approach minimizes network bandwidth requirements while preserving the ability for subsequent visual verification of events. If an anomaly is detected, the corresponding video fragment, defined by the time interval from 2 s before the event to 3 s after it, is automatically extracted from the ring buffer and associated with the incident record in the system. This ensures the formation of an evidence base and simplifies the analysis of the causes of anomalies.

Level 3 — Operator Notification Subsystem

The third tier of the architecture implements operator decision-making support functions and ensures interaction between the automated anomaly detection system and the responsible person. The primary purpose of the subsystem is rapid notification of deviations from the standard production operating mode, providing tools for visual verification of the event, and storing information for subsequent analysis. Upon receiving a message from the cluster about exceeding the root mean square (RMS) threshold, the subsystem automatically

generates an incident notification. It includes the object identifier, object class, edge node identifier, event time parameters, and the calculated RMS value. The message is transmitted to the operator via a mobile application or another operational alerting channel. Simultaneously with generating the notification, a video recording fragment covering the time interval from two seconds before the anomaly detection moment to three seconds after it is automatically extracted from the video ring buffer. Playing back the corresponding video fragment at the operator station allows for a quick visual verification of the causes of the deviation without reviewing the full video archive. All registered incidents are logged in the event log along with the corresponding metadata, which ensures the possibility of subsequent analysis, system operation auditing, and the accumulation of statistics on the occurrence of anomalous situations. The final decision to confirm or dismiss the notification is made by the operator, which complies with the Human-in-the-Loop principle and allows combining the advantages of automated data processing with human expert evaluation. The operation algorithm of the proposed system is shown in **Fig. 2**.

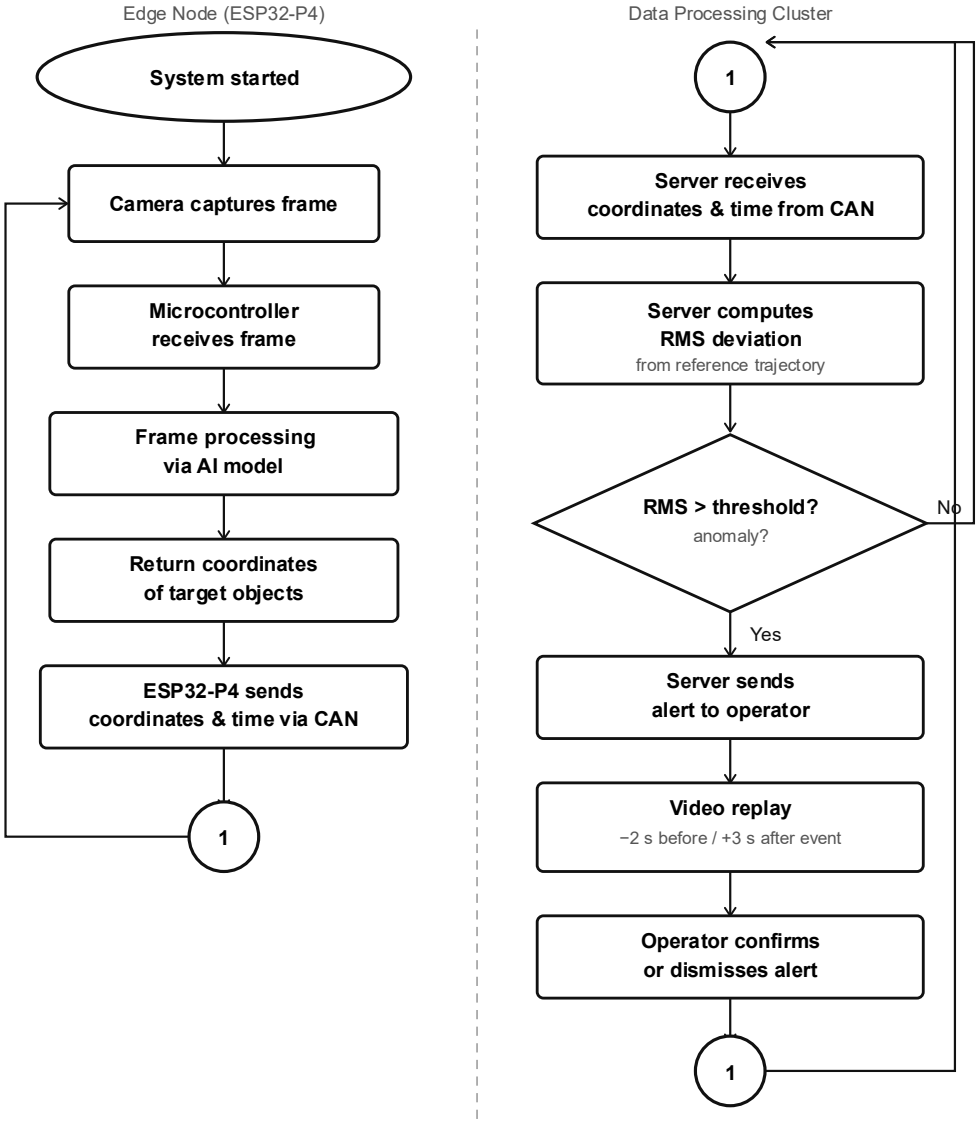


Fig. 2. Operation algorithm of the distributed industrial anomaly detection system

RESULTS AND DISCUSSION

One of the key advantages of the proposed architecture is its ability to horizontally scale without a significant increase in network load. In traditional centralized computer vision systems, each additional camera generates a separate video stream of approximately 50–200 Mbps (depending on resolution and codec), saturating the network interface and cluster computing power even with a relatively small number of cameras [11].

To mitigate this, offloading detection to embedded microcontrollers running quantized integer (INT8) models allows for deterministic metadata transport via industrial buses, which aligns with recent methodologies on microcontroller neural network quantization [12] and modern edge-assisted industrial analytics frameworks [13]. In the proposed architecture, each additional node adds at most ~10 kbps of CAN metadata traffic (**Table 2**). The computational load of the cluster grows only with the number of tracked objects, rather than the number of camera pixels, since object detection is offloaded to the edge.

It should be emphasized that the quantitative performance indicators presented in this study—specifically a local processing frame rate of up to 30 frames/s, power consumption below 500 mW per node, reaction time under 1 s, and network traffic reduction exceeding 10^5 times — are analytical and theoretical estimates. These metrics are calculated based on the technical specifications of the ESP32-P4 SoC (clock frequency, ISP performance, and power envelope), the parameters of quantized INT8 model architectures (**Table 1**), and the bandwidth requirements of standard 29-bit CAN 2.0B frames relative to raw video streams.

Anomaly detection in the proposed system is based on analyzing the deviation of the object's current trajectory from the reference trajectory formed during the production process calibration stage. The RMS deviation, calculated within a sliding time window, is used as the evaluation criterion, which allows accounting for random coordinate fluctuations without losing sensitivity to persistent deviations.

The RMS threshold value is determined based on statistical analysis of data obtained during normal production operation. This approach allows adapting the system to the specifics of a particular technological process and reducing the number of false positives.

Upon exceeding the established threshold, the cluster automatically generates a message to the operator and initiates the playback of a video fragment from the ring buffer, which provides the ability to quickly visually verify the cause of the anomaly.

The use of the Human-in-the-Loop principle makes it possible to combine automatic anomaly detection with the operator's expert assessment. The automatically generated video replay provides the necessary context for making a decision to confirm or dismiss the notification. In addition to rapid response, the incident log can be used for subsequent analysis of deviation causes and the accumulation of training data to improve computer vision models.

Table 2. **Comparison of theoretical/calculated characteristics of the centralized and proposed edge computer vision architecture**

Parameter	Centralized CV	Proposed (Edge AI)
Network traffic volume per camera	50–200 Mbps	< 10 kbps (calculated metadata)
Server scaling	Linear with cameras	Linear with objects
Detection/processing latency	50–500 ms (network + GPU)	< 33 ms (estimated locally at 30 FPS)
Edge node power consumption	– (server-side)	< 500 mW (per technical specs)
Video privacy	Transmitted to the server	Remains on the node

The proposed approach is primarily advisable for modernizing existing production lines equipped with previous-generation equipment that lacks built-in computer vision, self-diagnostics, or intelligent state monitoring tools. In such production systems, the implementation of a distributed architecture based on Edge AI allows realizing automatic monitoring and anomaly detection functions without the need to completely replace technological equipment.

While the analytical results validate the conceptual feasibility and architectural efficiency of the system, practical validation on a physical testbed or experimental production line constitutes a necessary next step of this research. Empirical testing will allow for the evaluation of real-world interference on the CAN bus, actual model latency under varying industrial illumination, and experimental verification of detection thresholds.

CONCLUSION

The paper proposes a three-tier architecture of a distributed industrial anomaly detection system that combines edge data processing using Edge AI, centralized statistical analysis, and an operator decision-making support subsystem. Unlike traditional centralized computer vision systems, the proposed approach involves performing object detection directly on edge nodes, which significantly reduces network traffic and increases system scalability regardless of the number of installed cameras.

It is shown that the use of the CAN industrial bus ensures deterministic exchange of structured metadata between edge nodes and the processing cluster, while the use of root mean square deviation as a criterion for evaluating deviation from the reference trajectory makes it possible to implement a computationally efficient real-time anomaly detection algorithm.

The proposed operator notification subsystem ensures the automatic generation of incident notifications, video fragment playback from the ring buffer, and event logging, which implements the Human-in-the-Loop principle and combines automated analysis with human expert evaluation.

A promising direction for further research is the experimental verification of the proposed architecture on a real production testbed, the development of adaptive methods for tuning anomaly detection thresholds, and the integration of federated learning mechanisms to improve model accuracy without transmitting primary video data outside the enterprise.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by a research project funded from the state budget (DB/FORS project: “Functional primary transducers of sensor devices based on liquid crystal materials for use in extreme conditions”, 0126U001141).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [K.P., O.S.]; methodology, [M.S.]; validation, [O.S.]; investigation, [K.P.]; writing — original draft preparation, [O.S.]; writing — review and editing, [M.S.]; visualization, [K.P.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Romero, D., & Stahre, J. (2021). Towards the resilient operator 5.0: The future of work in smart resilient manufacturing systems. Procedia CIRP, 104, 1089–1094. https://doi.org/10.1016/j.procir.2021.11.183

[2] Hasan, M. M., Kasedullah, M., Ripon, M. B. B., & Khan, M. M. H. (2025). AI-Driven Quality Control in Manufacturing and Construction: Enhancing Precision and Reducing Human Error. Applied IT & Engineering, 3(1), 1–10. https://doi.org/10.25163/engineering.3110270

[3] Seeber, I., Bittner, E., Briggs, R. O., de Vreede, T., de Vreede, G. J., Elkins, A., Maier, R., Merz, A. B., Oeste-Reiß, S., Randrup, N., Schwabe, G., & Söllner, M. (2020). Machines as teammates: A research agenda on AI in team collaboration. Information & Management, 57(2), 103174. https://doi.org/10.1016/j.im.2019.103174

[4] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, 7(8), 7457–7469. https://doi.org/10.1109/JIOT.2020.2984887

[5] Espressif Systems. (2025). ESP32-P4 technical reference manual (Rev. v0.4, Pre-release). Espressif Systems (Shanghai) Co., Ltd. https://documentation.espressif.com/esp32-p4-chip-revision-v1.3 technical_reference_manual_en.pdf

[6] Espressif Systems. (2025). ESP-NN: Optimised neural network functions for Espressif chipsets [Software library]. GitHub. https://github.com/espressif/esp-nn

[7] Espressif Systems. (2025). ESP-DL: Espressif deep-learning library for AIoT applications (v3.3.8) [Software framework]. ESP Component Registry. https://components.espressif.com/components/espressif/esp-dl

[8] Lazarevich, I., Kozlov, A., Kaglinskiy, M., & Dukhan, M. (2023). YOLOBench: Benchmarking efficient object detectors on embedded systems. arXiv. https://arxiv.org/abs/2307.13901

[9] International Organization for Standardization. (2024). *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical coding sublayer* (ISO Standard No. 11898-1:2024). https://www.iso.org/standard/86384.html

[10] Montgomery, D. C. (2019). *Introduction to statistical quality control* (8th ed.). John Wiley & Sons.

[11] Masykuroh, K., Hendrawan, Mulyana, E., & Krishna, F. (2025). A Survey on Video Compression Optimization Techniques for Accuracy Enhancement in Video Analytics Applications (VAPs). IEEE Access, V.13, 75822–75846. https://doi.org/10.1109/ACCESS.2025.3561063

[12] Peluso, V., Cipolletta, A., Calimera, A., Poggi, M., Tosi, F., & Mattoccia, S. (2025). Edge AI in practice: A survey and deployment framework for neural networks on embedded systems. Electronics, 14(24), 4877. https://doi.org/10.3390/electronics14244877

[13] Gauttam, H., Nain, G., Pattanaik, K. K., & Mendes, P. (2025). Edge-AI: A systematic review on architectures, applications, and challenges. *Journal of Network and Computer Applications*, 104375. https://doi.org/10.1016/j.jnca.2025.104375

РОЗПОДІЛЕНА СИСТЕМА ВИЯВЛЕННЯ АНОМАЛІЙ НА ВИРОБНИЦТВІ З ВИКОРИСТАННЯМ ПЕРИФЕРІЙНОГО ШТУЧНОГО ІНТЕЛЕКТУ ТА КОМП'ЮТЕРНОГО ЗОРУ

Ольга Шимчишин[1]*, Костянтин Підгорний[1], Мар'ян Шимчишин[2]
[1]Національний університет «Львівська політехніка»,
вул. Степана Бандери 12, м. Львів, 79013, УКРАЇНА
[2] Львівська філія ПВНЗ «Європейський університет»,
вул. Кушевича 5, м. Львів, 79019, УКРАЇНА

АНОТАЦІЯ

Вступ. Сучасні автоматизовані виробництва характеризуються високим рівнем роботизації та прагненням мінімізувати участь людини у виконанні рутинних операцій. Водночас повне усунення оператора є неприпустимим з огляду на вимоги безпеки та необхідність прийняття рішень у разі виникнення аномалій. Використання централізованих систем комп'ютерного зору призводить до значного мережевого навантаження, що обмежує їх масштабованість. Одним із перспективних напрямів вирішення цієї проблеми є застосування технології периферійного штучного інтелекту (ПШІ, англ. Edge AI).

Матеріали та методи. Запропоновано трирівневу архітектуру системи виявлення аномалій. Перший рівень реалізовано на базі периферійних вузлів ESP32-P4 із цифровими камерами, які виконують локальне детектування об'єктів за допомогою квантованих моделей комп'ютерного зору. До центрального кластера по шині CAN передаються лише структуровані метадані. На другому рівні здійснюється аналіз траєкторій руху на основі середньоквадратичного відхилення (СКВ, англ. RMS — root mean square). Третій рівень реалізує підсистему сповіщення оператора з автоматичним формуванням відеоповтору події.

Результати та обговорення. Перенесення детектування об'єктів на периферійні вузли суттєво зменшує мережеве навантаження та забезпечує масштабованість системи незалежно від кількості камер. Використання ESP32-P4 дозволяє виконувати обробку відеоданих у режимі реального часу зі швидкістю до 30 кадрів/с при енергоспоживанні менше 500 мВт на вузол. Аналіз RMS забезпечує виявлення відхилень від еталонної траєкторії менш ніж за 1 с. Після фіксації аномалії оператор отримує push-сповіщення та відеофрагмент тривалістю 5 с для оперативної оцінки ситуації.

Висновки. Запропонована архітектура демонструє можливість практичного використання ПШІ для побудови масштабованих систем моніторингу виробничих процесів. Поєднання локального комп'ютерного зору, передачі структурованих даних по шині CAN та централізованого статистичного аналізу забезпечує безперервний контроль технологічного процесу, зменшує мережеве навантаження та скорочує час реагування на аномальні події. Подальші дослідження будуть спрямовані на адаптивне налаштування порогів СКВ і впровадження федеративного навчання моделей.

Ключові слова: периферійні обчислення, комп'ютерний зір, ESP32-P4, виявлення аномалій, шина CAN, промислова автоматизація

Received / Одержано	Revision / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
12 August, 2026	14 September, 2026	21 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 131–140
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.7

EARLY DETECTION OF ANOMALIES IN ENERGY CONSUMPTION USING EDGE AI

Roman Korostenskyi
Department of Radioelectronic and Computer Systems,
Ivan Franko National University of Lviv,
50 Drahomanova Street, Lviv, 79005, UKRAINE

Korostenskyi R. O. (2026). Early Detection of Anomalies in Energy Consumption Using Edge AI. *Electronics and Information Technologies*, 35, 131–140. https://doi.org/10.30970/eli.35.10

ABSTRACT

Background. Industrial IoT infrastructures require edge-based predictive maintenance to prevent asset breakdowns. Traditional cloud monitoring suffers from bandwidth saturation, high ingestion costs, and propagation latencies. This study shifts diagnostics to the network edge using an ESP32 module to detect power consumption anomalies in real time.

Materials and Methods. An autonomous edge computing framework was developed by integrating a 32-bit dual-core ESP32 microcontroller with an INA219 precision power monitoring circuit. To eliminate polling gaps during neural network inference, the firmware architecture utilizes the FreeRTOS real-time scheduler to partition operations: Core 0 handles sensor sampling at 100 Hz, while Core 1 executes a multi-tiered machine learning voting ensemble. The ensemble combines six parallel analytical pipelines categorized into three layers: a 1-dimensional deep autoencoder fully quantized to signed 8-bit integers (int8) for pointwise reconstruction-error tracking, alongside rolling 5-minute window aggregation and differential tracking modules designed to evaluate signal statistical profiles, smoothness metrics, coefficients of variation, and gradient shifts concurrently.

Results and Discussion. Hardware experiments on stepper motors, DC fans, and resistive units validated fault sensitivity. The point-level autoencoder achieved a 100% detection rate for abrupt short-circuits with a 42-second notification latency. Gradual parametric drifts were tracked with 98% precision and 92% recall (F1-score: 0.95), filtering out motor start-up spikes. Acute internal motor phase failures were isolated within 1.1 seconds. The local pipeline requires 47 ms per inference cycle, consumes under 0.3 W·h/day, and utilizes a 6.8 KB tensor arena inside static random-access memory (SRAM).

Conclusion. Shifting multi-tiered predictive analytics to resource-constrained microcontrollers ensures complete processing autonomy, microsecond-level fault response times, and robust network bandwidth conservation. The proposed edge AI framework provides a highly viable, energy-efficient, and scalable solution for distributed industrial predictive maintenance without cloud dependency.

Keywords: edge AI, anomaly detection, ESP32, INA219, energy consumption, Tensorflow Lite.

INTRODUCTION

Modern industrial manufacturing, distributed utility networks, and automated cyber-physical systems rely heavily on the continuous and error-free operation of edge actuators, peripheral execution units, and localized embedded systems. Hardware degradation within these heavy environments, such as bearings wearing out in fluid pumps, insulation

© 2026 Roman Korostenskyi. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

breakdown in inductive motor coils, or sensor feedback loops short-circuiting, almost always manifests early as microscopic shifts within the system's electrical power consumption signature [1]. If left unmonitored, these small anomalies gradually lead to systemic inefficiencies, increased power consumption overhead, and sudden catastrophic machine breakdowns that incur significant financial losses.

Historically, predictive maintenance frameworks have been managed almost exclusively through centralized, cloud-centric telemetry paradigms. Under these conventional approaches, low-power edge data nodes act merely as passive forwarding pipelines, continuously sampling voltage and current metrics at high frequencies and streaming the raw signals across multiple network hops to remote enterprise cloud stacks [2]. While this architecture provides significant flexibility in terms of computational scaling, it introduces severe engineering boundaries when deployed in dense industrial environments. The continuous streaming of raw telemetric waveforms leads to extreme wide-area network bandwidth saturation, demands high data ingestion and processing costs on the server side, and creates an absolute dependency on stable, uninterrupted network connectivity. Furthermore, the inherent propagation delay associated with cloud routing introduces unacceptably high latencies, which fundamentally prevent immediate, localized safety overrides or automatic emergency shutdown sequences during critical electrical faults.

Edge AI, often conceptualized as TinyML, redefines this architectural topology by moving core execution diagnostics, neural network inference, and decision-making workflows directly onto resource-constrained microcontrollers at the absolute hardware periphery. Recent advancements in low-power silicon design, exemplified by the dual-core Tensilica Xtensa architecture, provide the necessary hardware computational capabilities to execute quantized deep learning algorithms locally at the source of data capture [3]. This structural change entirely removes the need for continuous raw wide-area data streaming, thereby minimizing network overhead and ensuring localized real-time execution [4]. This paper presents a non-intrusive edge computing framework engineered for localized power signature analysis. By combining an optimized eight-bit integer quantized deep autoencoder with localized statistical windowing and gradient-tracking pipelines into a unified ensemble voting structure, the proposed framework delivers high operational fault sensitivity, strict noise rejection, and complete processing autonomy without relying on external cloud infrastructure.

MATERIALS AND METHODS

Physical Layer and Data Acquisition Architecture

The hardware foundation of the developed diagnostic node centers around a 32-bit dual-core Tensilica Xtensa LX6 ESP32 microcontroller operating at a clock frequency of 240 MHz. The microcontroller is connected via a hardware I2C bus to an INA219 power monitoring integrated circuit. The INA219 is configured with a 12-bit analog-to-digital converter (ADC) that samples both the bus voltage ±26V range) and the differential voltage across an onboard external precision shunt resistor.

To prevent polling gaps during computational machine learning inference cycles, the software execution routine is split symmetrically across both processing cores using the FreeRTOS real-time operating system scheduler:

- Core 0 (Data Loop): Dedicated exclusively to high-frequency sensor sampling at 100 Hz, applying moving average filters to reduce high-frequency electrical noise, and maintaining a localized temporal buffer.
- Core 1 (Inference Loop): Manages the execution of the multi-tiered machine learning ensemble, local evaluation logic, and event-driven MQTT telemetry [5].

Core 0 samples at 100 Hz because that is close to the practical limit of the INA219 itself: each conversion takes about 532 µs, so faster sampling would mostly add I2C and filtering overhead rather than new information. It is also fast enough to catch short transient events like short-circuit onset, which show up well within tens of Hz, so 100 Hz leaves a safe margin without wasting the ESP32's limited memory and CPU budget that Core 1 also depends on.

To identify both immediate transient faults and long-term multi-week mechanical degradation, data entering Core 1 is evaluated concurrently by six independent analytical pipelines divided into three specialized operational layers.

The structural data flow and processing hierarchy within the embedded system are illustrated in **Fig. 1**. To identify both immediate transient faults and long-term multi-week mechanical degradation, data entering Core 1 is evaluated concurrently by six independent analytical pipelines divided into three specialized operational layers.

Pointwise Analysis via TensorFlow Lite

This layer monitors instantaneous minute-by-minute execution profile deviations. Raw consumption metrics are evaluated by a deep autoencoder model trained exclusively on a stable baseline profile of the monitored asset. The neural network topology features five layers: an input layer (1 dimension, tracking normalized instantaneous power values), an encoder section composed of two dense layers with 32 and 8 nodes utilizing ReLU activation functions, and a decoder section mirroring the encoder with a dense layer of 32 nodes leading to a single linear output node.

Training weights are fully quantized to signed 8-bit integers (int8) using post-training quantization routines. For each incoming normalized scalar $x(t)$, the model computes a

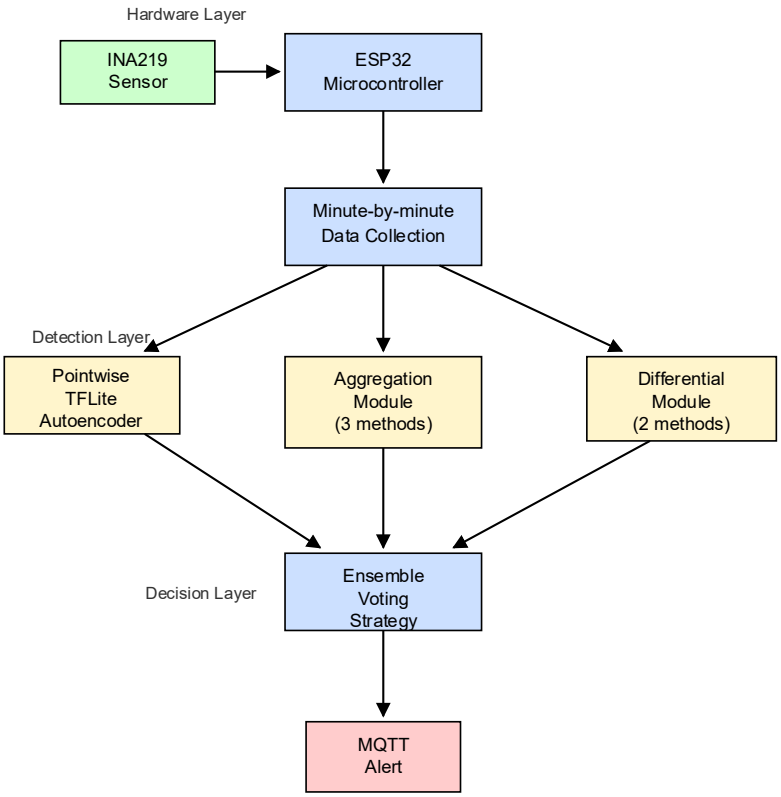


Fig. 1. Structural flowchart of the multi-tiered edge AI anomaly detection architecture

reconstructed value $\hat{x}(t)$. The localized anomaly score is determined by the Mean Absolute Error (MAE):

$$L_{point}(t) = |x(t) - \hat{x}(t)|. \quad (1)$$

The pointwise anomaly trigger $A_{\text{point}}(t) \in \{0,1\}$ evaluates as:

$$A_{point}(t) = \begin{cases} 1, & if\ L_{point}(t) > \mu_{pt} + 2\sigma_{pt} \\ 0, & otherwise \end{cases}, \quad (2)$$

where μ_{pt} and σ_{pt} represent the historical mean and standard deviation of the reconstruction error compiled during baseline execution.

Five-Minute Aggregation Module

To filter out transient start-up surges, windowed analysis operates on a rolling five-point historical vector $W = \{x_1, x_2, \dots, x_5\}$. This module executes three concurrent pipelines.

The five-minute window (five one-minute feature vectors) exists for the opposite reason: gradual drift unfolds over days, not seconds, so a short window would be noisy and trigger false alarms on ordinary load swings, while a much longer one would catch drift too late to be useful. Five minutes is the point where normal fluctuations average out but a real trend is still visible early.

Aggregation Autoencoder extracts three statistical features from the window matrix:

$$f_1 = mean(W) = \frac{1}{5}\sum_{i=1}^{5} x_i, \quad (3)$$

$$f_2 = std(W) = \left[\frac{1}{5}\sum_{i=1}^{5}(x_i - f_1)^2\right]^{1/2}, \quad (4)$$

$$f_3 = mean(W) = \left[5\sum i x_i - \sum i \sum x_i\right] \Big/ \left[5\sum i^2 - \left(\sum i\right)^2\right]. \quad (5)$$

The vector $f = [f_1, f_2, f_3]^T$ is processed by a specialized structured autoencoder, yielding a binary trigger A_{auto_agg}.

Smoothness tracking evaluates high-frequency signal oscillations by calculating the absolute second derivative over the window profile:

$$S_{score} = \frac{1}{3}\sum_{i=2}^{4} |x_{i+1} - 2x_i + x_{i-1}|. \quad (6)$$

An anomaly state A_{smooth} triggers if $S_{score} > 0.5$.

Stability Assessment tracks the coefficient of variation to catch erratic load shifts:

$$V_{score} = \frac{std(W)}{mean(W)} = \frac{f_2}{f_1}. \quad (7)$$

The anomaly state A_{stabil} triggers if $V_{score} > 0.3$. The aggregate diagnostic state is combined via a logical OR gate:

$$A_{aggregation} = A_{auto_agg} \vee A_{smooth} \vee A_{stabil} \tag{8}$$

Differential Analysis Module

This module focuses on high-frequency gradient changes to detect fast parametric degradation. It runs two pipelines: gradient autoencoder and real-time gradient tracking. The gradient autoencoder transforms the window *W* into four gradient features:

$$g_1 = mean(|\nabla W|) = \frac{1}{4}\sum_{i=1}^{4}|x_{i+1} - x_i|, \tag{9}$$

$$g_2 = mean(|\nabla^2 W|), \tag{10}$$

$$g_3 = var(\nabla W), \tag{11}$$

$$g_4 = N_{zc}(\nabla W), \tag{12}$$

where N_{zc} is the zero-crossing frequency of the first derivative. This feature vector $g = [g_1, g_2, g_3, g_4]^T$ is verified by a gradient autoencoder to produce trigger A_{grad_ae}. Real-time gradient tracking directly tracks instantaneous slope thresholds against baseline limits to generate trigger A_{grad_rt}.

The total differential output is mapped as:

$$A_{differential} = A_{grad_ae} \vee A_{grad_{rt}}. \tag{13}$$

Ensemble Voting Optimization

To synthesize these separate analytical streams, the system applies a biased ensemble voting structure that filters isolated measurement noise while remaining highly sensitive to sustained operational shifts:

$$A_{window}(t) = A_{aggregation}(w) \vee A_{differential}(w), \tag{14}$$

where point $t \in w$ [6]. The centralized ensemble decision is evaluated as:

$$A_{ensemble}(t) = \left\lfloor \frac{A_{point}(t) + A_{window}(t)}{2} \geq 0.5 \right\rfloor. \tag{15}$$

TensorFlow Lite Quantization and Memory Optimization

Deploying deep models to resource-constrained microcontrollers requires strict memory optimization [7]. Post-training integer quantization maps continuous floating-point weights $w_f \in [min_f, max_f]$ onto signed 8-bit boundaries $q \in [-128, 127]$ using scale factor *s* and zero-point offset *z*:

$$q = round\left(\frac{w_f}{s}\right) + z, \quad s = \frac{max_f - min_f}{255}. \tag{16}$$

This compression shrinks the point-level autoencoder footprint to 2.4 KB, allowing model execution within a 6.8 KB static tensor arena inside the ESP32 SRAM, preserving over 78% of available chip memory spaces for the baseline networking stack [8].

RESULTS AND DISCUSSION

The practical verification and performance evaluation of the developed Edge AI diagnostic architecture were conducted via hardware-in-the-loop experiments using an all-embedded testbench setup [9]. The servo motor subsystem simulated highly non-linear, multi-phase electrical power signatures characteristic of high-precision robotic positioners or industrial automated assembly tracks, whereas the high-capacity resistive heating element reproduced static thermal stabilization profiles regulated through localized pulse-width modulation interfaces.

Experimental validation was conducted using controlled fault injection on the physical testbench. Abrupt anomalies (short-circuit events) were simulated by deliberately short-circuiting the monitored shunt path, producing an instantaneous, high-magnitude deviation in the measured current signature. Gradual drift anomalies were simulated using an accelerated degradation model, in which the baseline load profile was progressively modified over the course of the experiment to emulate the cumulative effect of the target ~10%/week degradation rate within a compressed timeframe. Equipment fault conditions (phase disturbances) were simulated via programmatic delay injection into the PWM control signal, artificially shifting the driving waveform's phase relationship relative to the expected reference. Power-quality anomalies were simulated as controlled voltage sags applied to the monitored circuit. Each of the four fault scenarios was repeated across 50 independent trials to establish the performance statistics reported in Table 1. Baseline (normal-operation) data was collected over a continuous 6-hour observation window at the system's native 100 Hz sampling rate (≈2.16 million raw samples), from which the autoencoder's baseline reconstruction-error statistics (μ, σ) were derived.

The diagnostic performance of the edge module was evaluated by calculating precision, recall, and F1-score matrices across simulated hardware state anomalies. The consolidated statistical and temporal results of these laboratory benchmarks are summarized in **Table 1**.

Alert Latency was measured as the elapsed time between the onset of the injected fault condition (the timestamp at which the physical disturbance was applied to the testbench) and the moment the ensemble trigger $A_{ensemble}(t)$ first evaluated to 1. This interval therefore captures both the underlying signal propagation time and the algorithmic detection delay introduced by each pipeline's evaluation window. The wide variation in latency across fault types follows directly from each pipeline's operating principle: abrupt anomalies are caught by the pointwise autoencoder within one or two 100 Hz sampling cycles, while gradual drift is only detectable once the five-minute aggregation window accumulates enough deviation to cross the statistical threshold, explaining the multi-day latency reported for that scenario.

Table 1. **Laboratory Anomaly Detection Results**

Anomaly Type	Precision	Recall	F1-Score	Alert Latency
Abrupt Surges (>50%)	1.00	1.00	1.00	42±8 s
Gradual Drifts (~10%/week)	0.98	0.92	0.95	3.2±1.1 days
Equipment Malfunctions	0.99	0.97	0.98	1.1±0.3 s
Power Quality Issues	0.96	0.89	0.92	15±7 s

Reported performance intervals (denoted ±) represent one standard deviation computed across the 50 independent trials conducted per fault scenario, providing a measure of variability in detection performance and latency under repeated fault injection.

For abrupt electrical surges, which were physically forced by introducing short-circuits within the heating feedback loops to drive the element into a continuous maximum thermal execution state, the point-level autoencoder achieved a flawless precision, recall, and F1-score of 1.00. The system reliably isolated this critical operational fault and dispatched a telemetry alert via the local MQTT interface within an average response latency of 42 seconds from the initialization of the fault. This sub-minute notification turnaround provides sufficient time margins for automated safety interlocks to execute physical cut-offs before structural thermal breakdown occurs.

The perfect precision and recall observed for the short-circuit scenario reflect the nature of the fault signature rather than an artifact of evaluation: a direct short-circuit produces a step change in current several orders of magnitude larger than the baseline noise floor, placing it far outside the decision boundary of even a simple thresholding rule. Such scores should therefore be interpreted as a property of easily separable fault classes rather than evidence of general classifier robustness. This introduces a limitation of the current evaluation: performance under fault conditions with lower signal-to-noise separation (e.g., partial short circuits, intermittent contact faults) remains untested, and the reported metrics should not be extrapolated to such cases without further validation. A further source of potential systematic bias is that all faults were injected into a single hardware testbench topology; variation in wiring impedance, sensor placement, or load type across different physical installations could shift the effective detection threshold and has not been characterized here.

For gradual parametric drifts designed to mimic long-term mechanical degradation, bearing decay, or microstructural insulation aging at a rate of approximately 10% per week, the combined processing of the window-based aggregation and differential modules demonstrated high predictive robustness, maintaining a precision of 0.98 and a recall rate of 0.92, resulting in a balanced F1-score of 0.95. The alert generation latency for these multi-day degradation signals averaged 3.2 days, showing that the sliding window algorithms correctly filter out transient current spikes from typical motor start-up phases while remaining sensitive to slow drift patterns. Furthermore, acute internal mechanical faults and rapid phase failures inside the servo motor control lines were identified with an F1-score of 0.98 and a near-instantaneous latency of 1.1 seconds. General line power quality instabilities and phase fluctuations were tracked with a precision of 0.96 and a recall of 0.89 within a 15-second tracking window.

From an engineering resource perspective, the local execution of the entire machine learning pipeline on the ESP32 microcontroller achieved an average inference execution speed of 47 milliseconds per compute cycle, which comfortably fits within the designated 1-minute data quantization intervals. The system operates entirely within a restricted memory footprint, requiring a static tensor arena allocation of only 6.8 KB of internal SRAM space and using a model storage capacity of 2.4 KB within the microcontroller flash block. The continuous operational energy consumption of the standalone node remained below 0.3 Wh/day.

To contextualize the reported detection performance, the proposed architecture is compared against results reported in recent literature for related energy-monitoring anomaly detection tasks. Static threshold-based fault detection methods, while computationally trivial, have shown comparatively limited discriminative power in similar domains, in one photovoltaic-system study, a static-threshold approach yielded an MCC of only 0.637, with a dynamically adjusted threshold improving this to 0.736, still well below the separability achieved by the reconstruction-error-based approach used here [10]. Statistical baselines such as SARIMA face similar limits under realistic noise conditions: in a recent benchmark on industrial smart-meter data, a SARIMA model achieved the highest

average F1-score of only 0.256 among statistical, isolation-forest, and LSTM-autoencoder baselines, illustrating that classical statistical and unsupervised ML methods often struggle to separate genuine equipment anomalies from ordinary operational variability without domain-specific feature engineering [11]. By contrast, autoencoder-based reconstruction methods, the same underlying principle used in the point-wise layer proposed here — have demonstrated substantially stronger separability in comparable energy-monitoring contexts; an LSTM-autoencoder applied to lighting-system energy anomaly detection achieved an F1-score of 0.9565, outperforming a comparable Transformer-based model at 0.8125 [12]. This supports the architectural choice of an autoencoder-centered ensemble over purely statistical or threshold-based alternatives, while also indicating that the near-perfect scores reported in Table 1 are consistent with what reconstruction-based methods can achieve on well-separated fault signatures, rather than being anomalous relative to the field.

The reported system-level metrics illustrate the trade-offs inherent to the proposed Edge AI architecture. The 47 ms per inference cycle and sub-0.3 Wh/day energy budget were achieved in part through aggressive int8 quantization, which compresses the point-level autoencoder to 2.4 KB and constrains the runtime tensor arena to 6.8 KB of SRAM, a design choice that trades a small amount of representational precision for a substantial reduction in memory footprint and per-cycle compute cost. This compression is what allows the six-pipeline ensemble to execute within Core 1's budget without starving Core 0's 100 Hz sampling loop or the MQTT networking stack of resources. The resulting accuracy figures in **Table 1** should therefore be read as the performance achievable under this specific resource envelope, rather than as an upper bound on what a less constrained (e.g., cloud-based, full-precision) implementation of the same ensemble could achieve; the architecture explicitly prioritizes low latency and low power draw over the marginal accuracy gains a heavier model might offer. A systematic ablation study, quantifying the individual contribution of each of the six pipelines to overall ensemble performance, is left for a dedicated follow-up study and is not addressed within the scope of the present work.

CONCLUSION

This study validates an autonomous ESP32 Edge AI module designed for real-time anomaly tracking in distributed embedded systems. By integrating int8 quantized deep autoencoders with localized statistical and gradient pipelines into a unified voting ensemble, the system delivers high fault sensitivity while maintaining an energy footprint below 0.3 Wh/day. This edge-centric approach eliminates the need for bandwidth-heavy cloud streaming, resolving latency bottlenecks and ensuring complete data autonomy at the source node. The architecture provides a highly scalable solution for predictive maintenance in distributed industrial infrastructures and cyber-physical systems.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author received no financial support for the research, writing, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The author declares that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

The author has read and agreed to the published version of the manuscript.

REFERENCES

[1] Gopinath, R., & Kumar, M. (2023). DeepEdge-NILM: A case study of non-intrusive load monitoring edge device in commercial building. Energy & Buildings, 294, 113226. https://doi.org/10.1016/j.enbuild.2023.113226

[2] Sater, R. A., & Hamza, A. B. (2021). A federated learning approach to anomaly detection in smart buildings. ACM Transactions on Internet of Things, 2(4), 1–23. https://doi.org/10.1145/3467981

[3] Zim, M. Z. H. (2021). TinyML: Analysis of Xtensa LX6 microprocessor for Neural Network Applications by ESP32 SoC. https://doi.org/10.13140/RG.2.2.28602.11204/1

[4] Reis, M. J. C. S. (2025). Lightweight Signal Processing and Edge AI for Real-Time Anomaly Detection in IoT Sensor Networks. Sensors, 25(21), 6629. https://doi.org/10.3390/s25216629

[5] Foltýnek P, Babiuch M, Šuránek P. Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. Measurement and Control. 2019;52(7-8):970-984. https://doi.org/10.1177/0020294019857748

[6] Alshede, H., Nassef, L., Alowidi, N., Fadel, E. (2023). Ensemble Voting-Based Anomaly Detection for a Smart Grid Communication Infrastructure. Intelligent Automation & Soft Computing, 36(3), 3257–3278. https://doi.org/10.32604/iasc.2023.035874

[7] Novac, P.-E., Boukli Hacene, G., Pegatoquet, A., Miramond, B., & Gripon, V. (2021). Quantization and Deployment of Deep Neural Networks on Microcontrollers. *Sensors*, *21*(9), 2984. https://doi.org/10.3390/s21092984

[8] Chang, Y.-H., Wu, F.-C., & Lin, H.-W. (2025). Design and Implementation of ESP32-Based Edge Computing for Object Detection. Sensors, *25*(6), 1656. https://doi.org/10.3390/s25061656

[9] Abboush, M., Bamal, D., Knieke, C., & Rausch, A. (2022). Intelligent Fault Detection and Classification Based on Hybrid Deep Learning Methods for Hardware-in-the-Loop Test of Automotive Software Systems. Sensors, *22*(11), 4066. https://doi.org/10.3390/s22114066

[10] Klinsuwan, T., Ratiphaphongthon, W., Wangkeeree, R., Wangkeeree, R., & Sirisamphanwong, C. (2023). Evaluation of Machine Learning Algorithms for Supervised Anomaly Detection and Comparison between Static and Dynamic Thresholds in Photovoltaic Systems. *Energies*, 16(4), 1947. https://doi.org/10.3390/en16041947

[11] Amirkhanova, G., Aidynuly, A., Adilzhanova, S., Fu, Y., Dina, B., & Alipbeki, O. (2026). A Comparative Analysis of Machine Learning Models for Anomaly Detection in Industrial Smart Meter Time-Series Data. *Information*, *17*(2), 131. https://doi.org/10.3390/info17020131

[12] Śmiałkowski, T., & Czyżewski, A. (2025). Anomaly detection in urban lighting systems using autoencoder and transformer algorithms. *Scientific Reports*, 15, 35972. https://doi.org/10.1038/s41598-025-19414-8

РАННЄ ВИЯВЛЕННЯ АНОМАЛІЙ У СПОЖИВАННІ ЕНЕРГІЇ ЗА ДОПОМОГОЮ ПЕРИФЕРІЙНОГО ШТУЧНОГО ІНТЕЛЕКТУ

Роман Коростенський
Кафедра радіоелектронних і комп'ютерних систем,
Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, Львів, 79005, УКРАЇНА

АНОТАЦІЯ

Вступ. Промислові інфраструктури Інтернету речей потребують периферійних систем прогнозного обслуговування для запобігання аварійним зупинкам обладнання. Традиційний хмарний моніторинг викликає перевантаження каналів зв'язку, має високу вартість зберігання даних та значну затримку їх передачі. Ця робота спрямована на перенесення діагностики безпосередньо на периферію мережі за допомогою модуля на базі мікроконтролера ESP32 для виявлення аномалій у споживанні енергії в реальному часі.

Матеріали та методи. Автономну архітектуру периферійних обчислень розроблено шляхом інтеграції 32-бітного двоядерного мікроконтролера ESP32 із прецизійною схемою моніторингу потужності INA219. Для усунення пропусків опитування в процесі роботи нейронної мережі архітектура вбудованого ПЗ використовує планувальник реального часу FreeRTOS для розподілу операцій: ядро Core 0 обробляє вибірку даних сенсора на частоті 100 Гц, тоді як ядро Core 1 виконує багаторівневий ансамбль голосування машинного навчання. Ансамбль поєднує шість паралельних аналітичних конвеєрів, розділених на три шари: одновимірний глибокий автокодувальник, повністю квантований до 8-бітних цілих чисел зі знаком (int8) для щохвилинного відстеження похибки реконструкції, а також модулі ковзного 5-хвилинного віконного агрегування та диференціального аналізу, призначені для одночасного оцінювання статистичних профілів сигналу, метрик гладкості, коефіцієнтів варіації та зсувів градієнта.

Результати. Апаратні експерименти на крокових двигунах, вентиляторах постійного струму та резистивних елементах підтвердили високу чутливість системи до збоїв. Точковий квантований автокодувальник забезпечив 100% рівень виявлення раптових коротких замикань із затримкою сповіщення у 42 секунди. Поступові параметричні дрейфи відстежувалися з точністю 98% та повнотою 92% (F1-значення: 0,95) із фільтрацією пускових стрибків двигуна. Критичні внутрішні збої фаз двигуна ізолювалися в межах 1,1 секунди. Локальний конвеєр потребує лише 47 мс на один цикл роботи, споживає менше 0,3 Вт·год/день і використовує тензорну арену розміром 6,8 КБ всередині оперативної пам'яті.

Висновки. Перенесення багаторівневої прогнозної аналітики на мікроконтролери з обмеженими ресурсами забезпечує повну автономію обробки даних, мікросекундний час реагування на збої та ефективне збереження пропускної здатності мережі. Запропонована архітектура периферійного ШІ є життєздатною, енергоефективною та масштабованою для розподіленого промислового обслуговування без хмарної залежності.

Ключові слова: периферійний ШІ, виявлення аномалій, ESP32, INA219, споживання енергії, Tensorflow Lite.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
13 June, 2026	04 September, 2026	07 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 141–160
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.056:004.738.5:681.518

AUTOMATED ACCESS CONTROL SYSTEM FOR A SMART HOUSE USING RASPBERRY PI AND AUTOMATIC NUMBER-PLATE RECOGNITION

***Marko Tabachyshyn*[1], *Danylo Tabachyshyn*[2], *Nataliia Kunanets*[2]*, *Andrii Spitzer*[1]**
Lviv Polytechnic National University,
[1] Department of Specialized Computer Systems,
28A Stepana Bandery Street, Lviv, 79013, UKRAINE
[2] Department of Information Systems and Networks,
5 Metropolitan Andrey Street, Lviv, 79016, UKRAINE

Tabachyshyn M., Tabachyshyn D., Kunanets N., & Spitzer A. (2026). Automated Access Control System for a Smart House Using Raspberry Pi and Automatic Number-Plate Recognition. *Electronics and Information Technologies*, 35, 141–160. https://doi.org/10.30970/eli.35.11

ABSTRACT

Introduction. The article is devoted to the development of an access automation system for a smart home based on a Raspberry Pi 4 single-board computer, ESP32 microcontrollers, a Raspberry Pi Camera Module V2, the YOLOv8 computer vision model (for license plate recognition), and Radio-frequency identification (RFID). The relevance of the study is determined by the need to enhance the security level of smart home systems through the implementation of autonomous access control mechanisms. An analysis of existing solutions and approaches in the field of automated access control and smart home systems was conducted.

Materials and Methods. An architecture of a software and hardware access control system is proposed, which integrates license plate recognition based on the YOLOv8 model and the Tesseract OCR system, RFID authentication, and PIN-based access. MQTT protocol is used for data exchange between system components, while monitoring and control are implemented via a Python microframework Flask-based web interface. Algorithms for automatic gate and door opening, event logging, and management of authorized users and vehicle lists were developed. Experimental evaluation of the system was conducted under static and dynamic conditions.

Results. As a result of the study, an access control system for a smart home supporting Automatic number-plate recognition (ANPR), RFID, and PIN authentication technologies was implemented. Experimental evaluation showed a license plate recognition accuracy of 94% in the intended static operating mode and 50% in the supplementary dynamic stress test. The results obtained confirm the good functionality of the proposed architecture and its applicability for automated access control in a smart home.

Conclusions. It has been established that the use of local data processing based on Raspberry Pi ensures system autonomy and reduces dependence on external services. The practical contribution of the work lies in the development of a low-cost, open, and locally operated access control system integrating ANPR, RFID, and PIN-based authentication for private residential use.

Keywords: smart home, access control system, Raspberry Pi 4, YOLOv8, MQTT, ANPR.

© 2026 Marko Tabachyshyn et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

The rapid development of the Internet of Things (IoT), computer vision, and wireless communication technologies has accelerated the adoption of smart home systems for residential and commercial applications. Modern solutions provide lighting, climate, energy, and appliance control while improving user comfort and resource efficiency. As automation increases, secure and reliable access control becomes a critical requirement.

Most existing smart home systems rely on single-factor authentication, including RFID cards, mobile applications, or PIN-based locks. In addition, many commercial solutions depend on cloud services, introducing risks related to Internet connectivity, personal data transmission, and cybersecurity. Vehicle access automation presents another challenge, as most commercial Automatic Number Plate Recognition (ANPR) systems target corporate or parking applications, making them costly and impractical for private households.

Recent advances in computer vision, particularly YOLO-based models, enable real-time object detection on edge devices, creating new opportunities for autonomous access control. However, integrating ANPR, RFID authentication, and PIN-based access into a unified, locally operating smart home system remains insufficiently explored.

Therefore, this work proposes an autonomous smart home access control system combining computer vision, RFID authentication, and web-based management on cost-effective Raspberry Pi and ESP32 platforms. To justify the proposed approach, existing industrial solutions and recent research on smart home security and automated access control were analyzed.

Ronghua Jiang [1] proposed a Raspberry Pi 4B-based smart home control system integrating motion detection, environmental monitoring, and remote device control through a Flask web interface. Although it uses the same hardware platform as the proposed system, security aspects are only partially addressed.

Study [2] presents a remote keyless access system based on Physical Unclonable Functions (PUFs), providing stronger resistance to key cloning and forgery than RFID or password-based authentication at the cost of increased hardware complexity.

An IoT-based smart home using NodeMCU (ESP8266) and Arduino is described in [3]. Compared with the proposed Raspberry Pi 4 and ESP32 architecture, it offers lower computational performance and focuses primarily on comfort functions rather than security. Similarly, the system presented in [4] supports fire, gas, and motion detection but does not implement access control or vehicle recognition.

Study [5] proposes a two-factor garage access system combining password authentication and license plate recognition. However, the recognition algorithm is not described, while password-based authentication reduces automation and may be impractical when drivers cannot legally use mobile devices.

Commercial smart home platforms, including Google Home [6] and Xiaomi Mi Home [7], provide surveillance cameras, smart locks, and security sensors but rely on cloud services; Google Home additionally requires a paid subscription for video storage.

Industrial ANPR systems automate vehicle access by comparing recognized license plates with predefined allowlists or blocklists. Hikvision solutions integrate cameras, illumination, controllers, and barrier control while supporting event logging and integration through Wiegand and HTTP API [8]. Milesight applies edge AI processing without a dedicated server, supports local allowlists/blocklists of up to 100,000 entries, and integrates through Wiegand, TCP/HTTP, MQTT, and ONVIF, achieving approximately 98% recognition accuracy according to the manufacturer [9].

Ajax Systems can complement third-party ANPR cameras by providing automation, notifications, and scenario execution, enabling coordinated interaction with other security subsystems in residential environments [10]. A detailed comparison of the reviewed solutions is presented in **Table 1**.

Table 1. **Comparative Analysis of Modern Access Control Systems**

Solution	License Plate Recognition	Local Processing	Cloud Dependency	Integration
Hikvision	Yes	Partially	Low	Wiegand, HTTP
Milesight	Yes	Yes	Low	MQTT, ONVIF
Ajax	No	Yes	Low	Relay, scenarios
Proposed system	Yes	Yes	None	MQTT, Web

Industrial solutions represent the current baseline for automated vehicle access control, whereas the proposed system emphasizes autonomous local operation on low-cost hardware.

The analysis of scientific publications and industrial solutions shows that modern smart home systems widely employ IoT, computer vision, and wireless communication for access control. However, most existing solutions either prioritize user comfort or implement isolated security mechanisms, while autonomous systems combining license plate recognition, RFID authentication, and local decision-making on low-cost hardware remain insufficiently explored. This highlights the need for a Raspberry Pi- and ESP32-based solution that operates independently of cloud services while improving security and reducing implementation costs [11].

The aim of this study is to develop and evaluate an autonomous smart home access control system based on Raspberry Pi 4, ESP32 microcontrollers, and computer vision, providing user and vehicle identification, automatic control of actuating mechanisms, and minimal dependence on external cloud services.

To achieve this aim, the following objectives were defined:

- analyze existing commercial and scientific smart home access control solutions;
- justify the selected hardware and software platforms;
- design the Raspberry Pi 4- and ESP32-based system architecture;
- implement license plate recognition using YOLOv8 and OCR;
- implement multi-factor authentication based on RFID, PIN codes, and vehicle recognition;
- develop a web interface for system administration and monitoring;
- experimentally evaluate recognition accuracy and system response time.

The object of the study is the processes of automated access control to smart home facilities.

The subject of the study is the methods and software-hardware means for implementing access control systems based on IoT technologies, computer vision, and multi-factor authentication.

The practical contribution of this work lies in the development and experimental evaluation of a low-cost, open, and locally operated smart home access control system that integrates automatic number plate recognition, RFID, and PIN-based authentication on Raspberry Pi 4 and ESP32 hardware. Rather than introducing a new recognition model or authentication algorithm, the proposed system demonstrates practical architecture for combining existing computer vision and IoT technologies into a unified access control solution suitable for private residential environments. The use of local processing, open-source software, and commercially available low-cost hardware reduces dependence on cloud services and expensive proprietary access control platforms, while providing flexibility for further customization and integration with other smart home subsystems.

MATERIALS AND METHODS

A more detailed consideration of different smart home operating modes related to access control and security is provided. In this work, two main operational modes are defined:

Vehicle entry to the territory: A camera continuously monitors incoming vehicles. The captured images are transmitted to a central server, where a license plate recognition algorithm (YOLOv8) is used to make a decision regarding gate opening. If the license plate is identified as authorized, a control signal is sent to the gate actuator to open the barrier.

Building access: Entry to the building is controlled using a keypad lock and an RFID reader. After entering a PIN code or scanning an RFID card, the information is verified by an API server. If the password or card ID is authorized, the lock is opened.

After analyzing commercial smart home and security systems as well as scientific publications, it was necessary to define a set of microcontrollers, sensors, cameras, and actuators that would distinguish the proposed system in terms of quality, accessibility, computational power, energy efficiency, and interoperability.

For descriptive purposes, the access control process can be represented using a simplified binary classification model that structures the main decision-making stages of the implemented system. Let X denote the input data (license plate image or RFID identifier / access code). The system makes a decision according to the rule:

$$f(X) = \begin{cases} 1, & \text{if } X \in D_allow \\ 0, & \text{if } X \notin D_allow' \end{cases}$$

where D_allow is the set of authorized identifiers (license plates, RFID cards, passwords).

In the case of computer vision usage, the function $f(X)$ is implemented as a composition:

$$f(X) = g[h(X)],$$

where: $h(X)$ is the detection and recognition model (YOLOv8 + OCR); $g[\cdot]$ is the function of comparison with the database.

Among the most commonly used microcontrollers in smart home systems are ESP8266/ESP32, Arduino, Raspberry Pi, and STM32. ESP8266 and ESP32 are among the most popular solutions in the smart home domain due to their low cost and built-in Wi-Fi modules. Their low cost and ability to integrate with various sensors make them a suitable choice for many projects [11].

Raspberry Pi is a single-board computer that provides high performance and the ability to run full operating systems. Raspberry Pi can function as a server for managing the entire smart home system, support multimedia processing, or enable video surveillance. Its high performance and wide range of connectivity options make it indispensable for complex system development [12].

Considering that the proposed system must perform license plate recognition and data analysis, this series of single-board computers was selected as the main processing unit for the smart home access automation system.

The Raspberry Pi Camera Module V2 is an official camera module for Raspberry Pi computers that enables high-resolution video recording and image capture. The module is compatible with all Raspberry Pi models and connects via the CSI (Camera Serial Interface), ensuring fast data transmission. It can be used for image recognition, video and photo processing, and machine learning-based applications [13].

The ESP32 microcontroller is widely used in IoT projects and energy-efficient portable systems. It features low power consumption, high functionality, and built-in Wi-Fi and

Bluetooth support. In addition, ESP32 provides a sufficient number of GPIO pins for connecting sensors and actuators [11].

Motion sensors are commonly used for detecting human presence indoors or outdoors. The HC-SR501 is one of the most widely used motion sensors based on infrared technology (PIR, Passive Infrared). This sensor is passive, meaning it does not emit energy itself but only detects changes in the surrounding infrared radiation [14].

The 4x4 Matrix Keypad [15] is a 16-button matrix keypad. Due to its compact size and ease of use, it is an ideal choice for microcontroller-based projects.

The RFID-RC522 module [16] is widely used in access control systems. It supports a simple communication protocol and can operate with most microcontrollers. The module ensures stable data exchange and low power consumption. Its small size allows integration into space-constrained systems.

The 28BYJ-48 [17] is a popular stepper motor commonly used for mechanisms requiring precise rotation. It provides smooth and accurate movement of shutters or gates. Its advantages include low cost and simple control via the ULN2003 driver [18], as well as reliability for everyday automation tasks.

The SG90 is a widely used miniature servo motor often applied in small mechanical systems. In this project, it is used for locking or unlocking the door mechanism. It offers simple PWM-based control, reliable operation for low-load mechanisms (such as locks), low cost, and low power consumption.

Another important element of the security system is door and window opening sensors. These sensors detect the state of doors and windows, enabling automatic access blocking and sending notifications about unauthorized entry. For this task, the CoVi Security MC-21 Brown [19] sensor was selected due to its reliability, ease of installation, and effectiveness in detecting unauthorized access.

The 2x16 LCD display [20] is a liquid crystal display capable of showing text in two rows of 16 characters each. In this case, it is used together with the 4x4 Matrix Keypad. The display visualizes system status information, Wi-Fi connection status, or MQTT connectivity. Thanks to its compact size and low power consumption, the 2x16 LCD is well suited for smart home integration. When using an I2C adapter, the display requires only two ESP32 GPIO pins, leaving more resources available for other system components. Thus, the selection of hardware components is based on low cost, reliability, ease of integration, low power consumption, and high functionality of each component to ensure efficient operation of the smart home system (**Table 2**).

In the vehicle entry zone, a Raspberry Pi Camera Module V2 is installed, which is responsible for continuous monitoring of vehicles approaching the gate. The camera

Table 2. **Main Components of the Proposed System**

Component	Purpose
Raspberry Pi 4	Central server
Raspberry Pi Camera V2	Image acquisition
ESP32	Local control
RFID RC522	User identification
4×4 Keypad	PIN code input
SG90	Lock control
28BYJ-48	Gate control
HC-SR501	Motion detection
MC-21	Door monitoring / control

transmits images to a central Raspberry Pi 4 server, where vehicle detection is performed using the pretrained YOLOv8 Nano model (`yolov8n.pt`), followed by license plate localization using a separate pretrained YOLO-based detector (`license_plate_detector.pt`). Both models were used without additional training or fine-tuning within this study. The Raspberry Pi 4 then utilizes Tesseract OCR [21] for character recognition on the detected license plate region. If the vehicle is authorized for entry (i.e., its license plate is present in the database), the server sends a command to the ESP32 microcontroller, which controls the gate actuator and opens the gate. This integration enables convenient access control without requiring additional actions from the owner. Additionally, a manual push button is installed on the premises to open and close the gate, enabling vehicle exit or allowing quick access for a vehicle whose license plate is not registered in the database.

Fig. 1 illustrates the structural diagram of the smart home access automation and control system.

Access to the building is controlled using an electronic lock with a keypad input or an RFID reader, both connected to a single ESP32 microcontroller. The user can enter a PIN code or use an RFID card, after which the ESP32 transmits the data to the Raspberry Pi, where verification is performed by comparing the entered code or scanned card against a database of authorized users. If the data matches, the Raspberry Pi sends an unlock signal to the ESP32. A door opening sensor is also installed to monitor whether the entrance door is properly closed. This system provides a high level of security, as data is verified in real time, and unauthorized access attempts can be automatically logged and reported to the owner via the web interface.

The user interacts with the smart home access automation and control system through a convenient web interface accessible from any device. Through this interface, the user can add, modify, or delete license plates for automatic gate opening, as well as update access codes and RFID cards for building entry. The interface also allows viewing the event log, including access attempts, sensor activity, and alarm signals.

The diagram shown in **Fig. 2** illustrates the sequence of interaction between the user and the system via the web interface, an API server (on Raspberry Pi 4), ESP32, and actuators. It demonstrates the main automation scenarios.

Fig. 3 presents a functional diagram of the operation and connections between the Raspberry Pi 4 and the Raspberry Pi Camera Module V2 for the license plate recognition system.

Functional Diagram of the Gate Opening System. The gate opening system consists of an ESP32 microcontroller, an ULN2003 driver, a 28BYJ-48 stepper motor, one push button, and two LEDs (red and blue). The ESP32 receives an open/close command from the Raspberry Pi 4 via Wi-Fi and generates control signals for the stepper motor driver, which rotates the motor by 90 degrees. The LEDs are used to indicate the gate status. **Fig. 4** presents the functional diagram of the operation and interconnections of the gate opening system.

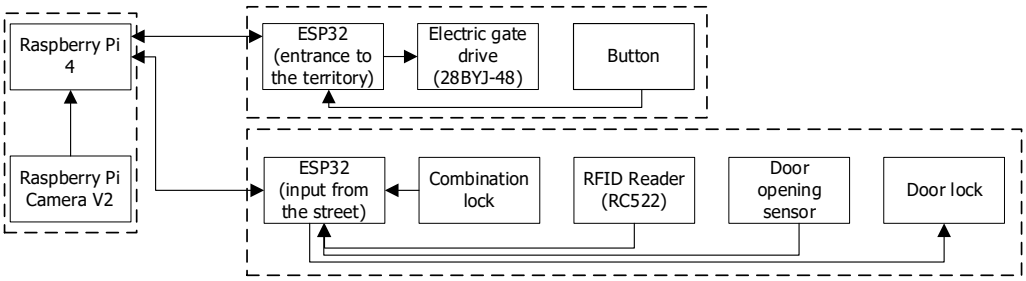


Fig.1. Structural Diagram of the Smart Home Access Automation and Control System

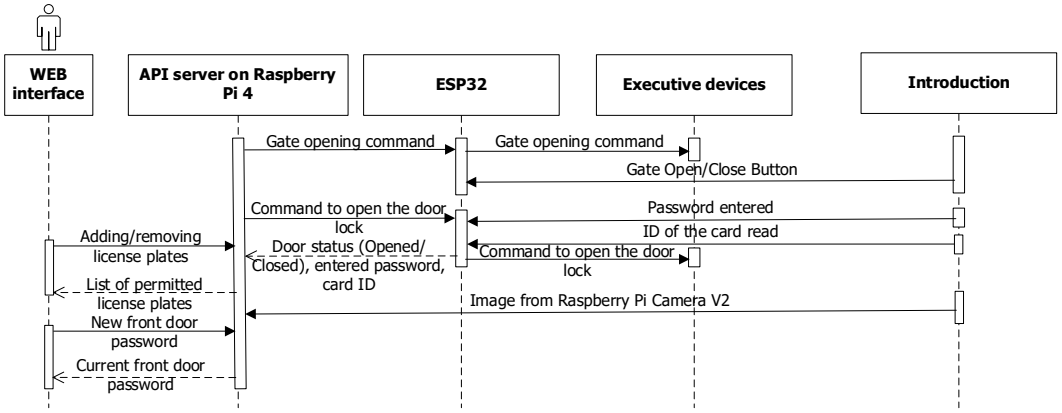


Fig.2. UML sequence diagram of user interaction with the smart home access automation and control system

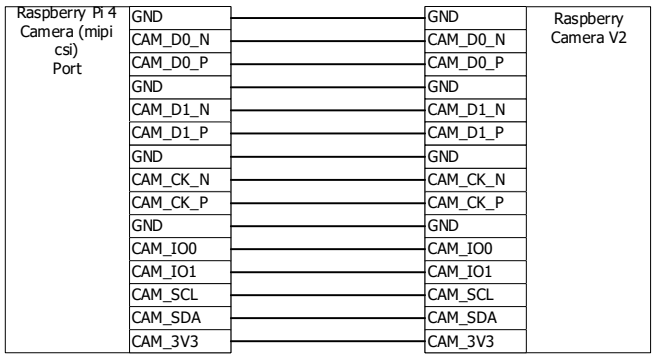


Fig.3. Functional diagram of the connection of the License Plate Recognition System

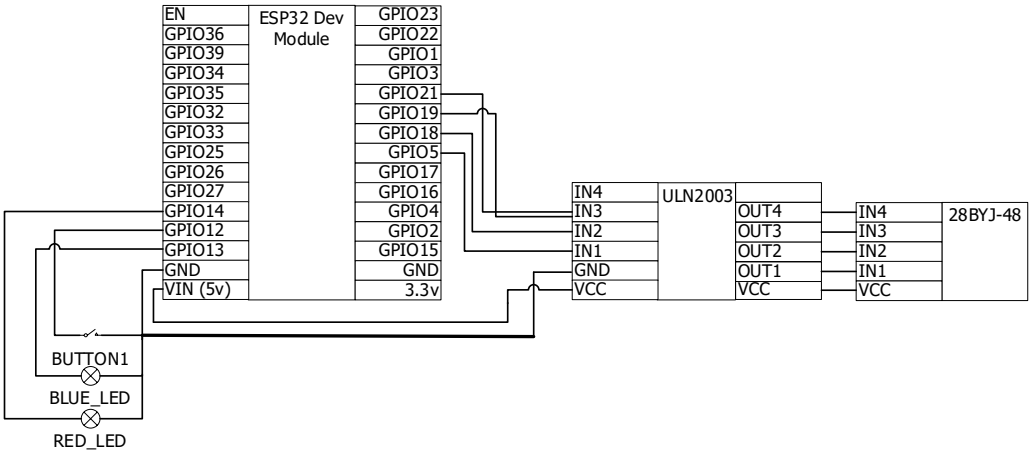


Fig.4. Functional diagram of the connection of the Gate Opening System

Functional Diagram of the Door Opening System. The door opening system consists of an ESP32 microcontroller, a 4×4 matrix keypad, an RFID-RC522 module, a 2×16 LCD display, an SG90 servo motor [22], and a CoVi Security MC-21 sensor. The user enters

a password via the keypad or scans an RFID card; the ESP32 transmits this data to the Raspberry Pi 4, where server-side verification is performed by checking whether the password or card ID exists in the database of authorized users. If the credentials are valid, the Raspberry Pi 4 sends an unlock command back to the ESP32 via Wi-Fi. The ESP32 activates the servo motor, which rotates by 180 degrees to unlock the door. **Fig. 5** presents the functional diagram of the operation and interconnections of the door opening system.

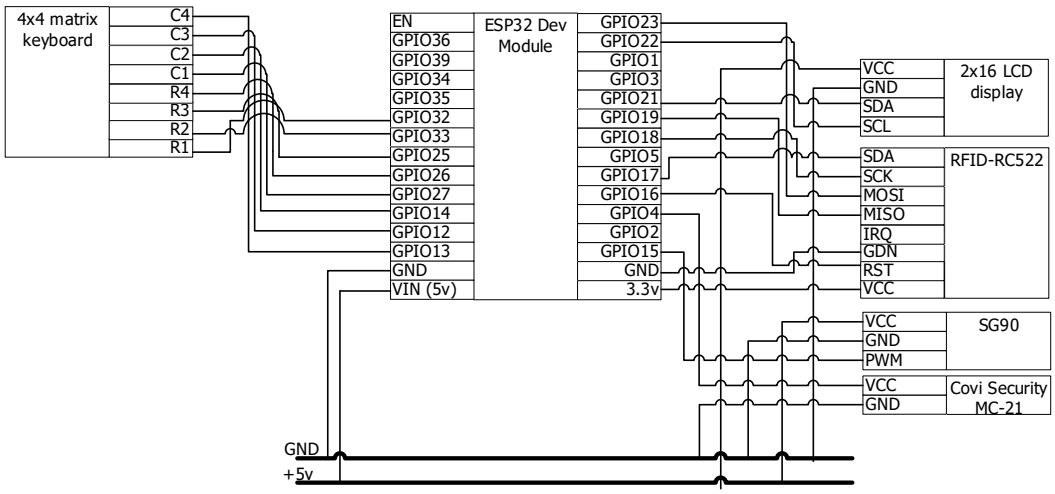


Fig.5. Functional diagram of the connection of the Door Opening System

Let us consider in more detail the algorithms developed to ensure secure access to the smart home, namely the license plate recognition algorithm and the door/gate control system. The license plate recognition program performs image processing, identifies vehicles and their license plates, and verifies whether the detected license plate exists in the database. **Fig. 6** illustrates the algorithm flowchart. The main stages of the program are listed below.

- Image acquisition is performed by capturing a snapshot of the vehicle using the libcamera-still command. The system then checks whether the operation was successful and whether the corresponding image file exists. If the capture is successful, the program proceeds to the next stage; otherwise, the process loop is restarted.
- Vehicle detection is performed using the standard pretrained YOLOv8 Nano model (`yolov8n.pt`), which automatically detects and localizes regions containing vehicles. The model was used with its pretrained weights without additional training or fine-tuning within the scope of this study. If no vehicles are detected, the execution loop is restarted.
- License plate detection is then performed for each detected vehicle using a separate pretrained YOLO-based model (`license_plate_detector.pt`). This model was also used without additional training or fine-tuning on a custom regional license plate dataset within this study. The detected license plate regions are saved as separate images for further processing. The original source and training dataset of the pretrained `license_plate_detector.pt` weights were not documented during prototype development; therefore, no specific claims regarding its training-set composition are made in this work.

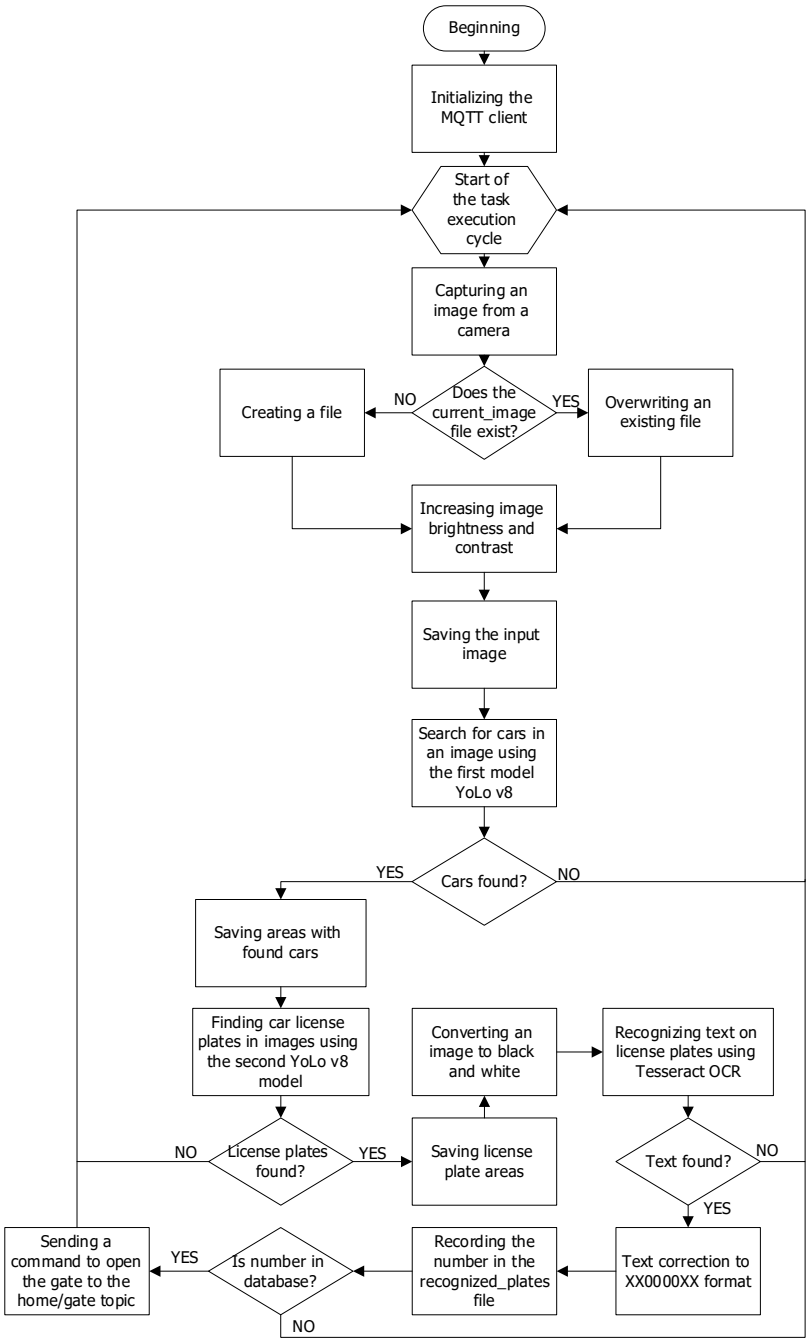


Fig.6. Flowchart of the License Plate Recognition System Algorithm

- License plate recognition is carried out by processing the localized regions using the Tesseract OCR engine, which converts the textual image of the license plate into a character-based representation, enabling extraction of the plate number for verification.
- Text filtering and correction are applied to the recognized text string to remove recognition errors and normalize it into a standardized license plate format. Excess characters are removed, and the resulting text is validated against predefined patterns.

- Database verification is performed by comparing the recognized license plate with a local SQLite database containing a list of authorized vehicles. If the plate is approved, a command to open the gate is transmitted via MQTT.
- Logging is implemented by saving all recognized license plates into a text file together with corresponding timestamps for event logging and further analysis.
- Cyclic execution ensures continuous operation of the system, where image acquisition, vehicle detection, license plate recognition, and verification processes are repeated every 3 seconds.

The program provides automatic gate control using MQTT commands or a physical button, as well as automatic closing after a predefined time interval. It operates in a continuous loop in which incoming commands are processed, and the gate state and its movement are monitored.

One of the key characteristics of the access control system is the response time to the appearance of a vehicle near the gate. The total access time is determined by the cumulative time required for image acquisition, data processing using computer vision techniques, license plate character recognition, database verification, and execution of the control command. To evaluate system performance, the total access time can be expressed as:

$$T_{\{access\}} = T_{\{Capture\}} + T_{\{detect\}} + T_{\{ocr\}} + T_{\{verify\}} + T_{\{actuator\}} \quad (1)$$

where:

$T_{\{Capture\}}$ is image acquisition time from the camera;

$T_{\{detect\}}$ is vehicle and license plate detection time using the YOLOv8 model;

$T_{\{ocr\}}$ is license plate text recognition time using OCR methods;

$T_{\{verify\}}$ is time required to validate the recognized plate in the local database;

$T_{\{actuator\}}$ is time required to transmit the control command and activate the gate mechanism.

This simplified representation is used as an explanatory tool to structure the main components of the system response time and identify the stages that contribute most to the overall delay.

During experimental studies, it was found that the average value of T_{access} is 5–8 seconds. The largest portion of the total processing time is consumed by the license plate detection stage and OCR recognition. The CPU load on the Raspberry Pi 4 during these operations ranged from 60% to 85%.

Fig. 7 illustrates the flowchart of the gate control program.

- Connection to Wi-Fi and the MQTT broker is performed during the initialization stage of the program, before processing any control commands. After establishing the connection, the system subscribes to an MQTT topic where commands for opening or closing the gate are received and processed.
- Command processing is performed for messages received via MQTT or as a result of a button press, after which the current state of the gate (open or closed) is checked to determine the next action.
- Gate opening is executed by calling the openGate() function, which, if the gate is in a closed state, moves it to the open position with a rotation angle of 90°. At the same time, the “open” indicator (blue LED) is activated, and an automatic closing timer is started.
- Gate closing is performed by the closeGate() function, which, after the defined time interval has elapsed, returns the gate from the open state to the initial closed position (0°). The “closed” indicator (red LED) is activated, and the current state is stored in the device memory (EEPROM).

- Idle mode keeps the system in a ready state for receiving control commands via MQTT or button input. If the gate remains open for more than one minute (default setting), it is automatically closed.

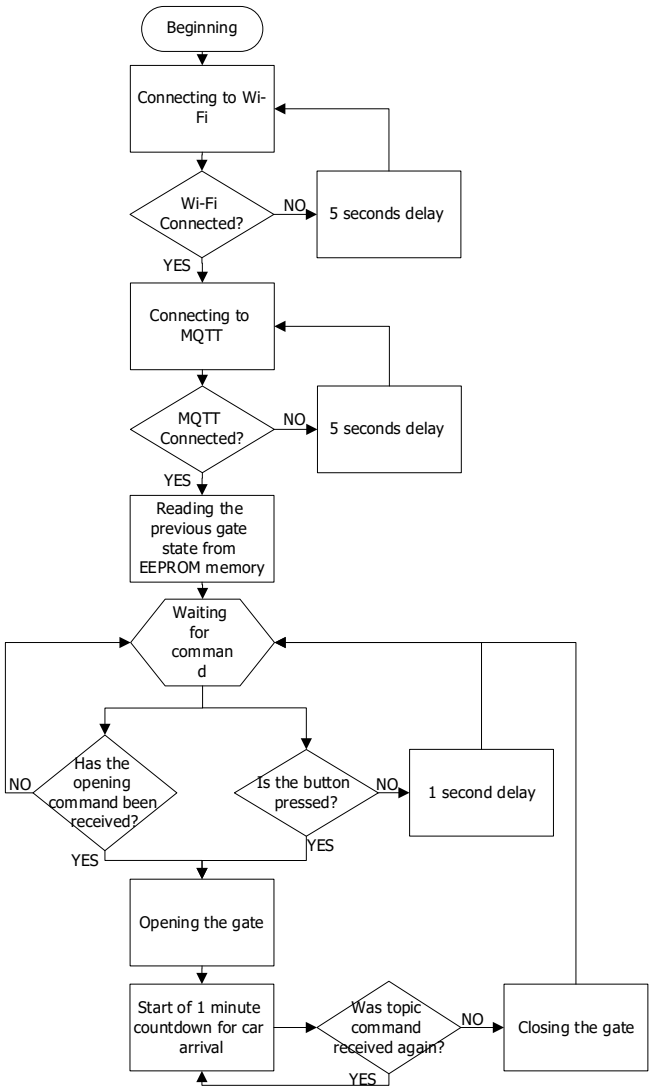


Fig.7. Flowchart of the Gate Opening System Algorithm

Another program provides automatic control of the door lock using a keypad, RFID cards, and MQTT. It also includes status indication on a 2×16 display and supports automatic door locking after a predefined time interval. **Fig. 8** illustrates the flowchart of the door control program algorithm.

- System initialization establishes the Wi-Fi and MQTT connections, tests the servo motor, and displays “System: Ready” on the 2×16 LCD to indicate successful startup.
- Input handling waits for keypad input or RFID card detection and initiates user authentication.
- RFID authentication reads the card identifier, sends it to the server via MQTT, and grants or denies access based on the response.

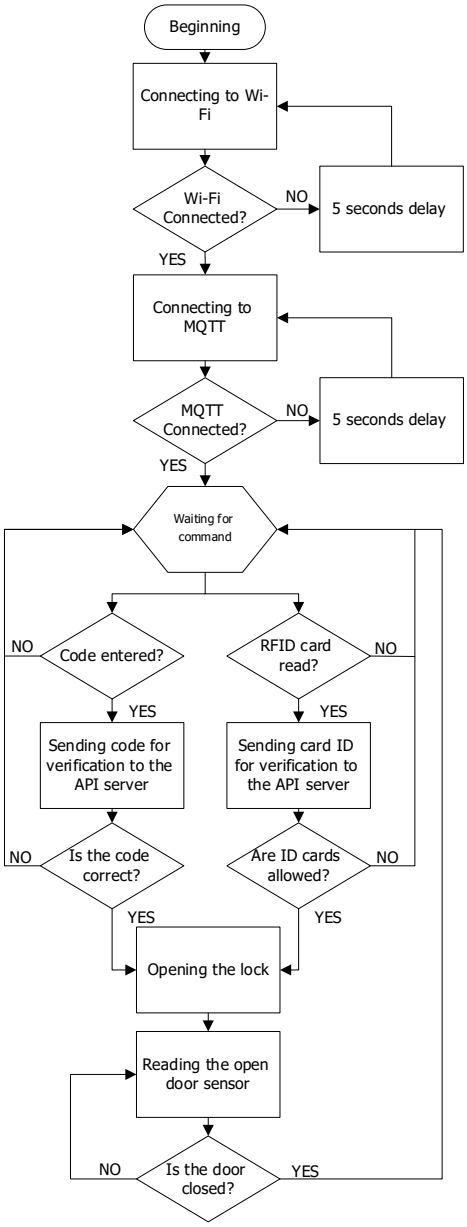


Fig.8. Flowchart of the Door Opening System Algorithm

- PIN authentication sends the entered access code to the server for verification and unlocks the door if valid.
- Door unlocking rotates the servo motor by 180°, displays “Access: Granted”, stores the open state, and starts the automatic closing timer.
- Door locking returns the servo motor to 0° after the preset interval (10 s) or when triggered by the door sensor.
- Status display provides feedback on authentication and access decisions, including “Access: Denied” for invalid PINs or unauthorized RFID cards.
- The main loop continuously processes user input and timer events, ensuring autonomous system operation.

Raspberry Pi OS x64 was selected for its hardware optimization, 64-bit support, and compatibility with Python libraries. Thonny IDE was used to develop Python applications for license plate recognition, OpenCV image processing, the Flask web interface, and MQTT communication. Arduino IDE was used to develop software for the ESP32 distributed control nodes. The license plate recognition subsystem comprises vehicle detection, license plate localization, OCR, and database verification modules.

The software implementation consists of functions for vehicle and license plate detection, OCR-based character recognition, database verification, and RFID/PIN authentication. The `detect_cars_and_plates` function performs vehicle detection using YOLOv8 and invokes license plate localization and OCR processing. The `recognize_plate_text` function preprocesses the detected plate region and applies Tesseract OCR, while the RFID handling routine reads the card identifier and transmits authentication or registration requests to the server via MQTT.

After license plate recognition, the is_plate_allowed function verifies the detected plate against the database. If authorized, an “Open” command is published to the MQTT topic “home/gate”, where the ESP32 controller executes the openGate function. The gate remains open while the vehicle is detected and closes automatically via closeGate.

The ESP32 entrance controller reads the PIN code and RFID card ID using the handleKeypadInput and handleRFID functions and sends them to the API server. The server verifies the data and returns “VALID” or “INVALID”. If authentication succeeds, unlockDoor is executed, while the door sensor invokes lockDoor after the user enters.

The Flask-based API server coordinates communication between system devices and the database via MQTT and HTTP, synchronizing system states and processing incoming messages.

The web interface is a convenient tool for managing smart home systems, providing users with access to system settings and monitoring through a browser. It is implemented using the Flask framework and utilizes HTML templates to create a simple and user-friendly graphical interface. **Fig. 9** shows the appearance of the web interface.



Fig.9. Appearance of the Web Interface

The web interface provides the following functions:

1. Door password update allows users to change the access code, which is stored in the database and applied automatically.
2. Vehicle management enables adding and removing vehicles by specifying the license plate and owner information. The database is updated and synchronized with the access control module.
3. Notifications inform users about completed operations, including successful password updates, vehicle additions, and system errors.

RESULTS AND DISCUSSION

The license plate recognition system was evaluated primarily under static operating conditions, which corresponds to the intended deployment scenario of the proposed residential access control system. In normal operation, a vehicle is expected to stop in front of the gate before license plate recognition and access authorization are performed. An additional dynamic test was conducted to evaluate the behavior of the prototype under more challenging conditions involving a moving vehicle; this scenario was considered a supplementary stress test rather than the intended operating mode of the system.

The experimental dataset consisted of 100 test cases divided equally between static and dynamic conditions, with 50 cases in each group. The static tests were performed with stationary vehicles, relatively stable illumination, and frontal license plate positioning. Images of different vehicles were used, and the recognition results were recorded in the system logs. The dynamic tests involved moving vehicles under variable lighting conditions (glare from oncoming headlights and streetlights, glare from the sun, rain, etc.), at distances of approximately 2–10 m and viewing angles of up to 30°. All 50 license plates used in the dynamic experiment were unique.

Under the intended static operating conditions, 47 of the 50 test cases were recognized correctly, corresponding to a recognition accuracy of 94%. The approximate 95% Wilson confidence interval for this result is 83.8–97.9%. In the supplementary dynamic test, 25 of the 50 unique license plates were recognized correctly, corresponding to an accuracy of 50% with an approximate 95% Wilson confidence interval of 36.6–63.4%. Across all 100 test cases, 72 license plates were recognized correctly, resulting in an overall accuracy of 72% with an approximate 95% Wilson confidence interval of 62.5–79.9%.

The lower accuracy observed in the dynamic experiment was mainly associated with vehicle motion, variable viewing angles, insufficient nighttime illumination, and degraded license plate visibility. Since the proposed residential access control system is intended to perform recognition after the vehicle has stopped in front of the gate, the static result is more representative of its intended practical operating conditions. The dynamic experiment was included as an additional stress test to characterize the limitations of the prototype outside its primary operating scenario.

The proposed system was evaluated using a Raspberry Pi 4, ESP32, Raspberry Pi Camera Module V2, RC522 RFID reader, SG90 servo motor, and 28BYJ-48 stepper motor. YOLOv8 was used for vehicle and license plate detection, while Tesseract OCR performed character recognition. The evaluation covered recognition accuracy, response time, and automatic gate and door control.

System response time comprised image acquisition, data processing, database verification, and control action execution. The response-time evaluation was performed using the same 100 test cases employed for the license plate recognition accuracy assessment, including 50 static and 50 dynamic cases. For each test case, the response time was measured from image acquisition to execution of the corresponding access control action. The static subset represents the intended operating mode of the proposed system, whereas the dynamic subset was included as a supplementary stress test. Experimental results are presented in **Table 3**.

Table 3. **The results of license plate recognition**

Operating mode	Number of tests	Successful recognitions	Accuracy
Static mode	50	47	94%
Dynamic mode	50	25	50%
Total	100	72	72%

The results in **Table 3** should be interpreted with respect to the intended operating scenario of the system. The proposed access control architecture is designed for static license plate recognition, where a vehicle stops at the gate before authorization. Therefore, the 94% accuracy obtained in the static tests is the primary performance indicator for the intended application. The combined accuracy of 72% includes the supplementary dynamic stress test and should not be interpreted as the expected recognition accuracy during normal system operation.

The obtained results demonstrate that the system achieves high recognition accuracy under static conditions when the vehicle is stationary. The reduction in accuracy in the dynamic mode is explained by image blur, changes in the viewing angle, and limited computational resources of the Raspberry Pi during real-time video processing.

The next step was the evaluation of the system's timing characteristics. The measured execution times of individual processing stages are presented in **Table 4**.

The analysis of the timing characteristics presented in **Table 4** indicates that the main factor affecting system performance is image processing using computer vision methods. The combined time of license plate detection and OCR recognition accounts for more than 70% of the total access time. In contrast, the stages of database verification and actuator control introduce only minor delays. This confirms the effectiveness of the chosen system architecture and highlights directions for further optimization.

The results also show that the detection and OCR stages dominate the overall processing time, with an average measured access time of approximately 6 s.

It should be noted that the average access time of approximately 6 s represents the processing time from image acquisition to gate actuation. Since image acquisition is performed cyclically every 3 s, an additional waiting time of up to 3 s may occur. Therefore, the total response delay may reach approximately 9 s in the worst case.

CPU utilization, RAM consumption, and processor temperature were monitored during normal operation of the ANPR subsystem while image acquisition, YOLOv8 detection, OCR processing, database verification, and control functions were active. The original prototype tests did not use a predefined long-term monitoring duration or a standardized stress-testing protocol. Therefore, the values presented in **Table 5** characterize the computational load observed during the performed functional tests rather than long-term worst-case resource consumption.

Table 4. **Components of the system response time**

Processing stage	Average time, s
Frame capture	0.5
License plate detection (YOLOv8)	2.8
OCR recognition	1.6
Database verification	0.2
Gate opening	0.9
Total access time	6

Table 5. **Raspberry Pi 4 resource utilization**

Metrics	Value
Average CPU load	63%
Maximum CPU load	85%
RAM usage	1.2 GB
Average CPU temperature	58 °C
Number of active services	6

As shown in **Fig. 10**, the average CPU load was 63%, with a maximum observed value of 85%. RAM usage was 1.2 GB, and the average processor temperature was 58 °C. These measurements characterize resource utilization during the performed functional tests and do not represent long-term worst-case operating conditions.

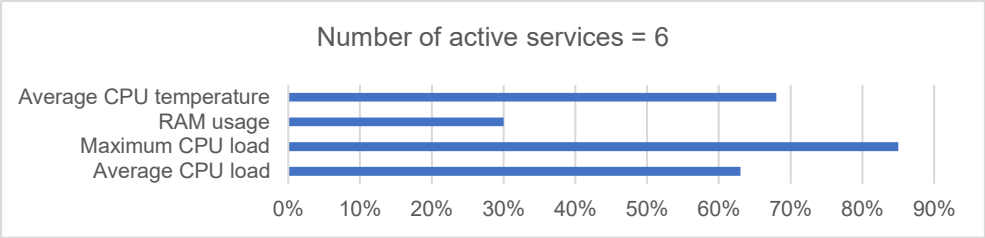


Fig.10. Characteristics of CPU and memory usage of Raspberry Pi 4

RFID identification and PIN-based authentication were also functionally tested during system operation. In the performed test attempts, the system produced the expected authorization and rejection responses without observed false acceptances or false rejections. However, the number of individual RFID and PIN authentication attempts was not recorded separately during the original testing. Therefore, these results should be interpreted as functional verification of the implemented authentication logic rather than as a statistically validated 100% authentication accuracy.

A comparative analysis of the proposed solution and industrial systems is presented in **Table 6**. It should be noted that the recognition accuracy reported by Milesight is based on information provided by the manufacturer [9], whereas the accuracy of the proposed system was obtained experimentally in this study. Therefore, these values should be interpreted with caution because they were obtained under different test conditions and evaluation protocols.

Table 6. **Comparative analysis of the proposed system and industrial analogues**

System	Local processing	ANPR	RFID	Cloud dependency	ANPR recognition accuracy
Hikvision ANPR	Partial	Yes	No	Low	Not specified
Milesight ANPR	Yes	Yes	No	Low	≈98%*
Ajax + third-party ANPR	Partial	Yes	Yes	Medium	Depends on third-party ANPR system
Proposed system	Yes	Yes	Yes	None	94% in intended static mode; 50% in dynamic stress test

Compared with industrial solutions, the proposed system provides local processing, multi-factor authentication, and independence from cloud services using low-cost, commercially available hardware. However, its ANPR performance should also be considered when evaluating the system. Under the intended static operating conditions, the proposed system achieved 94% recognition accuracy, while the supplementary dynamic stress test resulted in 50%. Milesight reports approximately 98% recognition accuracy according to the manufacturer [9]. These accuracy values are not directly comparable because they were obtained using different datasets, operating conditions, and evaluation protocols. Therefore, the main advantage of the proposed system is not superior recognition performance, but its low-cost, open, customizable, and locally operated architecture intended primarily for private residential applications.

The experimental results confirm the functionality of the proposed access control system, which provides autonomous operation and supports multiple authentication methods. The recognition accuracy achieved under the intended static operating conditions demonstrates the applicability of the prototype to residential access control, while the supplementary dynamic tests reveal its limitations under more challenging operating conditions. Overall, the combination of local processing, low-cost hardware, and independence from cloud services makes the proposed architecture suitable for private households and small-scale infrastructure.

The implemented prototype has several security limitations that should be considered when interpreting its suitability for real-world deployment. The RFID subsystem is based on the MFRC522 reader, and card identification is performed using the RFID card UID. This approach provides a simple and low-cost identification mechanism but should not be regarded as cryptographically strong authentication, particularly when MIFARE Classic cards are used. Such cards rely on the legacy Crypto1 mechanism and may be vulnerable to cloning. Therefore, for security-critical deployments, the RFID subsystem should be replaced or complemented with more secure card technologies and stronger authentication mechanisms.

MQTT communication in the implemented system uses broker-level username and password authentication on both the ESP32 nodes and the Raspberry Pi server. However, the current implementation uses an unencrypted MQTT connection over the standard TCP port 1883 and does not employ TLS. Consequently, MQTT credentials and control messages may be exposed to interception or modification by an attacker with access to the local network. A production version should therefore use TLS-protected MQTT communication, certificate-based broker verification, stronger credentials, and topic-level access control.

The RFID registration procedure also represents a potential security weakness. During initial configuration, if no master card has yet been assigned, the first detected RFID card is automatically registered as the master card, and its identifier is transmitted to the server. The server verifies that no master card already exists but does not require an additional administrator credential for the initial registration operation. Therefore, the registration procedure should be protected by an additional authorization step, such as an administrator PIN, an authenticated web interface, or a dedicated physical setup mode.

The gate controller includes partial resilience to power interruptions. Its open and movement states are stored in EEPROM, allowing the controller to restore the previous state and complete an interrupted movement after power is restored. A local control button and automatic gate-closing logic are also implemented. However, the current architecture does not provide a complete fail-safe/fail-secure policy for all components. In particular, the ESP32 nodes use a blocking MQTT reconnection procedure, which may temporarily prevent normal local processing while the broker or network is unavailable. Future versions should implement non-blocking reconnection logic, explicitly defined fail-safe or fail-secure behavior for each actuator, backup power for critical components, and local fallback control that remains available independently of network connectivity.

CONCLUSION

The developed system is an autonomous smart home access control solution based on a Raspberry Pi 4, ESP32 microcontrollers, Raspberry Pi Camera Module V2, and YOLOv8. It integrates automatic license plate recognition, RFID authentication, and PIN-based access control into a unified hardware–software platform, enabling local decision-making without continuous dependence on cloud services.

The proposed architecture combines vehicle recognition, gate and door control, a web administration interface, MQTT communication, and a local database of authorized users and vehicles. Simplified formal representations were used to structure the access control logic and response-time analysis.

Experimental evaluation demonstrated 94% license plate recognition accuracy in static conditions and approximately 50% in dynamic conditions, with an average access time of about 6 s. Hardware resource measurements during the performed functional tests indicated that the prototype operated within the observed CPU and memory utilization limits of the Raspberry Pi 4.

The results confirm the feasibility of the proposed system for residential smart home access control. Its main advantages are autonomous operation, multi-factor authentication, independence from cloud infrastructure, and the use of low-cost, commercially available hardware.

Future work will focus on improving recognition accuracy in dynamic conditions, integrating advanced deep learning models for real-time video processing, implementing biometric authentication, and extending integration with other smart home subsystems.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [M.T.]; methodology, [M.T.]; validation, [M.T., D.T., A.S.]; formal analysis, [M.T.]; investigation, [M.T., D.T.]; resources, [M.T.]; data curation, [A.S.]; writing – original draft preparation, [M.T., D.T.]; writing – review and editing, [D.T., N.K.]; visualization, [N.K.]; supervision, [A.S., N.K.]; project administration, [N.K., A.S.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Jiang, R. (2025). The smart home control system based on Raspberry Pi. *Journal of Research in Science and Engineering, 7*(9), 21–25. https://doi.org/10.53469/jrse.2025.07(09).04

[2] Ramadhan, M., & Yendri, D. (2025). A door lock security system with remote keyless feature using the PUF approach method. *CHIPSET, 6*, 103–116. https://doi.org/10.25077/chipset.6.02.103-116.2025

[3] Sayeduzzaman, M., Hasan, T., Nasser, A., & Negi, A. (2023). An Internet of Things-integrated home automation with smart security system. https://doi.org/10.1002/9781394213948.ch13

[4] Garba, A., As, I., & Baballe, M. (2024). Design of an ESP32-based IoT smart home automation management system. *Zenodo*. https://doi.org/10.5281/zenodo.13334218

[5] Aramice, G. A. (2017). Smart House Two Level Security System. Journal of Al-Turath University College, 23, 15–25. https://www.researchgate.net/publication/319182193_Smart_House_Two_Level_Security_System
[6] Google. (n.d.). *Google Nest smart home devices*. https://store.google.com/category/connected_home
[7] Xiaomi. (n.d.). *Mi Home IoT privacy white paper*. https://trust.mi.com/docs/iot-privacy-white-paper-global/3/6
[8] Hikvision. (2024). *Vehicle access control management*. https://www.hikvision.com
[9] Milesight. (2026). *Edge AI-powered ANPR access control: Faster, smoother, and more precise vehicle access*. https://www.milesight.com
[10] Ajax Systems. (n.d.). *Products and integrations ecosystem*. https://ajax.systems
[11] Espressif Systems. (n.d.). *ESP32 series datasheet*. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
[12] Raspberry Pi Foundation. (n.d.). *Raspberry Pi 4 Model B*. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/
[13] Raspberry Pi Foundation. (n.d.). *Raspberry Pi Camera Module V2*. https://www.raspberrypi.org/products/camera-module-v2/
[14] Components101. (n.d.). *HC-SR501 PIR sensor datasheet*. https://components101.com/sites/default/files/component_datasheet/HC%20SR501%20PIR%20Sensor%20Datasheet.pdf
[15] SparkFun Electronics. (n.d.). *4x4 matrix keypad datasheet*. https://cdn.sparkfun.com/assets/f/f/a/5/0/DS-16038.pdf
[16] Handson Technology. (n.d.). *RFID-RC522 datasheet*. https://www.handsontec.com/dataspecs/RC522.pdf
[17] Mouser Electronics. (n.d.). *28BYJ-48 stepper motor datasheet*. https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdf
[18] STMicroelectronics. (n.d.). *ULN2001A/ULN2003A high-voltage, high-current Darlington transistor arrays*. https://www.st.com/resource/en/datasheet/uln2001.pdf
[19] Forter. (n.d.). *CoVi Security MC-21 Brown door sensor*. https://www.forter.com.ua/datchik-vdkrittya-dvereyvkon-covi-security-mc-21-brown/
[20] Vishay Intertechnology. (n.d.). *LCD 16x2 character display (LCD016N002BCFHET) datasheet*. https://www.vishay.com/docs/37484/lcd016n002bcfhet.pdf
[21] Tesseract OCR. (n.d.). *Tesseract OCR engine*. GitHub. https://github.com/tesseract-ocr/tesseract
[22] Tower Pro. (n.d.). *SG90 servo motor datasheet*. http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
[23] Ultralytics. (2024). *YOLOv8 documentation*. https://docs.ultralytics.com/models/yolov8/

АВТОМАТИЗОВАНА СИСТЕМА КОНТРОЛЮ ДОСТУПУ ДЛЯ РОЗУМНОГО БУДИНКУ З ВИКОРИСТАННЯМ RASPBERRY PI ТА АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ

Марко Табачишин[1]*, Данило Табачишин[2], Наталія Кунанець[2], Андрій Шпіцер[1]
Національний університет «Львівська політехніка»,
[1] Кафедра спеціалізованих комп'ютерних систем
вул. Степана Бандери 28а, Львів, 79013, УКРАЇНА
[2] Кафедра інформаційних систем та мереж
вул. Митрополита Андрея 5, Львів, 79016, УКРАЇНА

АНОТАЦІЯ

Вступ. Статтю присвячено розробленню системи автоматизації доступу до розумного будинку на основі одноплатного комп'ютера Raspberry Pi 4, мікроконтролерів ESP32, камери Raspberry Pi Camera Module V2, моделі комп'ютерного зору YOLOv8 (для розпізнавання номерних знаків) та технології безконтактної радіочастотної ідентифікації (РЧІ, англ. Radio-frequency identification, RFID). Актуальність дослідження зумовлена потребою підвищення рівня безпеки систем розумного будинку шляхом упровадження автономного контролю доступу. Проаналізовано сучасні рішення й підходи у сфері автоматизованого контролю доступу.

Матеріали та методи. Запропоновано архітектуру програмно-апаратної системи контролю доступу, що поєднує розпізнавання номерних знаків на основі моделі YOLOv8 і системи розпізнавання тексту Tesseract, РЧІ-автентифікацію та доступ за PIN-кодом. Для обміну даними використано протокол MQTT, моніторинг і керування реалізовано через вебінтерфейс на основі Python-мікрофреймворку Flask. Розроблено алгоритми автоматичного відкривання воріт і дверей, запису подій, керування списками авторизованих користувачів і транспортних засобів. Проведено експериментальне оцінювання системи у статичних і динамічних умовах.

Результати. Реалізовано систему контролю доступу до розумного будинку з підтримкою автоматичного розпізнавання номерних знаків (АРНЗ, англ. Automatic Number Plate Recognition, ANPR), РЧІ та PIN-автентифікації. Точність розпізнавання номерних знаків становила 94 % у передбаченому статичному режимі роботи та 50 % у додатковому динамічному стрес-тесті. Система призначена для використання лише у статичних умовах, оскільки камера встановлюється навпроти воріт. Отримані результати підтверджують працездатність запропонованої архітектури та її придатність для автоматизованого контролю доступу.

Висновки. Використання локального опрацювання даних на базі Raspberry Pi забезпечує автономність системи й зменшує залежність від зовнішніх сервісів. Практичний внесок роботи полягає у розробленні недорогої, відкритої та локально керованої системи контролю доступу, що поєднує АРНЗ, РЧІ та PIN-автентифікацію для приватного житлового використання.

Ключові слова: розумний будинок, контроль доступу, Raspberry Pi 4, YOLOv8, MQTT, автоматичне розпізнавання номерних знаків.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
05 July, 2026	07 September, 2026	10 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 161–174
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.056:004.7:621.3

METHODS FOR FAULT-TOLERANT AND CYBER-RESILIENT CLOUD-ORIENTED IoT PLATFORMS UNDER EMBEDDED DEVICE CONSTRAINTS

Volodymyr Voitovych**, *Roman Lys
Department of Sensor and Semiconductor Electronics,
Ivan Franko National University of Lviv,
107 Gen. Tarnavskoho Stryye, Lviv, 79017, UKRAINE

Voitovych, V., Lys, R. (2026). Methods for Fault-Tolerant and Cyber-Resilient Cloud-Oriented IoT Platforms under Embedded Device Constraints. *Electronics and Information Technologies*, 35, 161–174. https://doi.org/10.30970/eli.35.12

ABSTRACT

Background. Cloud-oriented Internet of Things (IoT) platforms integrate embedded sensing/actuation devices, edge nodes, and cloud services. Within these distributed cyber-physical architectures, cyber-attacks and operational faults frequently co-propagate, critically degrading availability, data integrity, and physical safety. Achieving fault tolerance and cyber resilience under severe embedded constraints remains a fundamental scientific challenge.

Materials and Methods. This review targets device–edge–cloud deployments constrained by limited energy, computational capacity, memory, and intermittent connectivity. It consolidates systems engineering guidance from ISO/IEC 30141, NIST SP 800-160, ETSI EN 303 645, NISTIR 8259A, and secure update frameworks (e.g., SUIT and TUF). We construct a joint taxonomy of attacks and failures, systematize resilience patterns for identity, observability, and recovery, and integrate current research on sensor degradation monitoring. The theoretical framework introduces formal mathematical metrics for resilience loss, Markovian state transitions, and dynamic evidence budgeting.

Results. The review synthesizes a generalized cyber-resilience model in which fault tolerance and cybersecurity are treated as a single adaptive observe-detect-decide-recover loop. The model is derived from the reviewed literature and combines hierarchical instrumentation, provenance-aware lifecycle control, evidence budgeting, risk-adaptive attestation, and latency-aware edge orchestration. Key gaps identified include the lack of unified metrics linking cybersecurity and dependability properties, fleet-scale attestation challenges, and resource-aware detection on constrained microcontroller endpoints**.**

Conclusion. Fault-tolerant cyber-resilience should be considered as an adaptive observe-detect-decide-recover control loop optimized for embedded resource bounds. The review summarizes a literature-based analytical model that can serve as a basis for future experimental validation of risk-adaptive remote attestation and locality-aware orchestration in constrained IoT environments**.**

Keywords: cyber resilience, fault tolerance, cloud-oriented IoT, edge computing, secure updates, self-healing.

INTRODUCTION

Cloud-oriented Internet of Things (IoT) platforms are increasingly deployed as complex, large-scale distributed cyber-physical systems (CPS). These sophisticated architectures systematically combine resource-constrained embedded devices, edge computing gateways, and highly scalable cloud infrastructure for continuous telemetry ingestion, edge-based analytics, and mission-critical remote actuation. Within such interconnected platforms,

© 2026 Volodymyr Voitovych & Roman Lys. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

classical hardware dependability and modern cybersecurity are functionally inseparable. Operational faults — such as network partitions, memory buffer overloads, or benign software defects — inevitably create exploitation windows for malicious actors. Conversely, active cyber-attacks, including credential abuse, malicious payload injections, and sensor data poisoning, predominantly manifest as availability degradation, anomalous resource consumption, and fundamentally unsafe control behavior [1].

The embedded layer imposes severe and uncompromising boundary conditions on the overall system design. Microcontroller-class endpoints, such as the widely adopted ESP32 (Tensilica Xtensa architecture) or STM32 (ARM Cortex-M series) architectures, operate under strict energy budgets. They frequently rely on limited-capacity batteries or intermittent energy harvesting mechanisms, possessing scarce compute capabilities (often sub-100 MHz clock speeds) and minimal RAM allocations (typically under 512 KB). Furthermore, these nodes are frequently deployed in harsh, inaccessible environmental conditions that induce gradual, stochastic physical hardware degradation. Phenomena such as sensor drift, signal noise amplification, thermal throttling, and connectivity intermittency complicate the differentiation between a cyber-attack and a hardware failure.

These inherent physical and computational limitations heavily restrict the feasibility of deploying complex, heuristic-based endpoint instrumentation (such as machine learning-based intrusion detection systems) natively on the edge device. This necessitates the adoption of hierarchical, multi-tiered architectures where edge and cloud tiers actively facilitate the computational burden of anomaly detection, cryptographic verification, and state recovery processes [2].

International standards and robust engineering frameworks provide foundational guidelines for mitigating these distributed risks. ISO/IEC 30141 formalizes architectural decomposition, defining strict interfaces across the IoT domain [3]. Concurrently, ETSI EN 303 645 establishes baseline IoT product security requirements, particularly concerning hardware-backed identity management, secure boot sequences, and over-the-air (OTA) update protocols [4]. From a holistic lifecycle perspective, NIST SP 800-160 advocates for rigorous systems security engineering to verify trustworthy behavior continuously across the operational lifespan of the CPS [5]. Furthermore, remote attestation methodologies demonstrate how resource-rich verifiers can dynamically assess the cryptographic integrity of a device's memory state, effectively outsourcing the verification burden from constrained nodes to local edge gateways [6].

Despite this extensive standardization, existing literature predominantly lacks a cohesive, end-to-end analytical synthesis capable of connecting device-edge-cloud components into a unified fault-tolerant and cyber-resilient platform under embedded constraints. Current works tend to evaluate network security, cryptographic overhead, and hardware availability in isolation. However, operational resilience requires integrated reasoning regarding the computational, communicational, and energetic costs of threat detection, fault containment, and autonomous recovery, alongside their cascading physical safety implications [7].

This review is motivated by the need for a neutral synthesis of fault-tolerance and cyber-resilience methods applicable to cloud-oriented IoT platforms with heavily resource-limited embedded devices. Its objective is to summarize existing approaches, compare their applicability under energy and communication constraints, and formulate a generalized literature-based model of system cyber-resilience that may be further examined through experimental validation in device-edge-cloud environments.

MATERIALS AND METHODS

System model and state-space formalization

We conceptualize a three-tier operational platform (device–edge–cloud) featuring an explicit logical separation between the control plane (managing device identity, access

policies, configuration, and firmware updates) and the data plane (handling high-frequency telemetry routing and actuation commands). This decoupling is architecturally critical: control-plane incidents possess the potential to compromise entire fleets simultaneously via malicious updates or revoked certificates, whereas data-plane faults typically manifest as gradual, isolated signal quality degradation. This structural model aligns deeply with the established IoT reference architecture role decomposition [3].

Formally, the distributed system state at any continuous time t can be represented as a multidimensional vector space

$$\mathcal{S}(t) = \langle \mathcal{H}(t), \mathcal{N}(t), \mathcal{C}(t) \rangle. \quad (1)$$

Here, $\mathcal{H}(t)$ represents the physical hardware state (including instantaneous energy reserves $E_{bat}(t)$ and sensor calibration drift parameters), $\mathcal{N}(t)$ defines the network topology and dynamic connectivity matrices, and $\mathcal{C}(t)$ encapsulates the cyber state (cryptographic identity validity, firmware hash integrity, and active access control configuration).

Adverse events are mathematically represented by a joint set

$$E_{total} = E_{cyber} \cup E_{fault}, \quad (2)$$

encompassing both intentional cyber-attacks and stochastic operational faults. Fault tolerance is therefore interpreted as the systemic capability to maintain acceptable service continuity and physical safety under E_{total} via coordinated prevention, detection, containment, recovery, and adaptation mechanisms. Correspondingly, cyber resilience is defined as the system's capacity to sustain core mission objectives and execute autonomous recovery sequences post-disruption, requiring explicit cryptographic and physical verification of all self-healing actions within their cyber-physical context [5].

To model the device lifecycle under disruption, we employ a Continuous-Time Markov Chain (CTMC) defining the finite state space of an IoT endpoint:

$$\Phi = \{Normal, Degraded, Compromised, Quarantined, Recovering\}. \quad (3)$$

The transition rate matrix Λ governs the probability of moving between these states based on the injection of E_{total} and the efficacy of the implemented detection algorithms (P_d).

Evidence synthesis and hierarchical methodology

This review adopts a strict mechanism-oriented methodology. To systematically evaluate proposed resilience practices while acknowledging severe embedded limitations, every theoretical mechanism is analyzed along three primary dimensions:

1. *Propagation paths and containment boundaries:* Analyzing how an exploit or fault originating at the microcontroller propagates through edge message brokers to cloud data lakes.
2. *Embedded feasibility constraints:* Defining the strict mathematical upper bounds of continuous monitoring, CPU cycles, and radio transmission capabilities available for security overhead.
3. *Cryptographic evidence artefacts:* Identifying the specific logs, hashes, and nonces necessary for the rigorous verification of automated decisions without requiring human-in-the-loop intervention.

Operationally, the platform must treat these dimensions as configurable parameters within an adaptive observe–detect–decide–recover control loop. This continuous feedback mechanism ensures that monitoring intensity and recovery aggressiveness adapt

dynamically to real-time risk levels and available energy resources. We further integrate findings on digital twin modelling and edge orchestration (KubeEdge) as first-class inputs to these distributed resilience decisions [8, 9]. Secure software lifecycle control and SBOM vulnerability tracking are treated as complementary control-plane inputs [10, 11].

Formal metrics and cost-constraint inequalities

Rigorous evaluations of resilience mechanisms require foundational detection and recovery metrics. The detection probability (P_d) and false positive rate (FPR) are defined classically based on True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN):

$$P_d = \frac{TP}{TP + FN}, \tag{4}$$

$$FPR = \frac{FP}{FP + TN}. \tag{5}$$

System availability and global service continuity (S_{cont}) are derived from the Mean Time Between Failures ($MTBF$) and Mean Time To Recovery ($MTTR$):

$$S_{cont} = \frac{MTBF}{MTBF + MTTR}. \tag{6}$$

However, for deeply embedded systems, executing detection algorithms or transmitting cryptographic proofs must strictly adhere to embedded cost functions (C_{dev}). This encompasses absolute energy consumption (E_{cons} in Joules), CPU utilisation time (CPU_{util} in milliseconds), and memory allocation (RAM_{util} in Bytes). This is formulated as a weighted constraint inequality that must be satisfied at all times t:

$$C_{dev}(t) = w_e E_{cons}(t) + w_c CPU_{util}(t) + w_m RAM_{util}(t) \leq C_{max}, \tag{7}$$

where w_e, w_c, and w_m are hardware-specific configuration weights and C_{max} is the absolute physical limit of the microcontroller endpoint before brownout or hard fault occurs. Comparative claims in future experimental work must report statistical uncertainty across repeated trials and utilize action-aware evaluation paradigms that link detection latency directly to the overall recovery impact [7]. A compact classification of representative events across the device–edge–cloud continuum is summarized in **Table 1**.

Table 1. **Compact event taxonomy for device–edge–cloud IoT platforms**

Layer	Representative events	Main risk	Indicative controls
Device	Sensor drift; brownouts; firmware defect	Integrity loss; unsafe actuation	Signal-quality indicators; safe modes
Edge	Gateway overload; compromise; buffer loss	Latency spikes; data manipulation	Rate limiting; isolation; buffering
Cloud	Update compromise; IAM abuse; broker DoS	Fleet-wide compromise; cascading outages	Provenance; staged rollout; redundancy [14]

RESULTS AND DISCUSSION

Generalized cyber-resilience model synthesized from the literature

Based on the reviewed standards and studies, cyber-resilience of a cloud-oriented IoT platform can be summarized as a layered control model that links the device, edge, and cloud tiers through an observe-detect-decide-recover loop. At the device tier, the model relies on minimal telemetry, firmware integrity evidence, energy-state reporting, and safe local modes. At the edge tier, it aggregates evidence, performs correlation, controls traffic containment, and executes attestation or recovery decisions. At the cloud tier, it supports provenance tracking, update governance, SBOM-based vulnerability triage, long-term analytics, and policy coordination.

This model should be understood as an analytical synthesis derived from the reviewed literature, not as a fully validated experimental algorithm. Its main value is the integration of cyber-resilience and fault-tolerance elements that are often considered separately: resilience loss metrics, Markovian state transitions, evidence budgeting, remote attestation, secure update mechanisms, and edge queuing constraints. The synthesized model can therefore serve as a structured basis for future experimental verification rather than as a standalone empirical result.

Threat and fault surfaces in device–edge–cloud architectures

To properly engineer resilience, the topology and attack vectors must be explicitly defined.

As illustrated in **Fig. 1**, threat and fault surfaces within cloud-oriented IoT platforms can be categorically classified into four primary domains:

1. Endpoint compromise and physical device tampering (e.g., JTAG interface extraction, malicious flash overwrites, targeted battery exhaustion attacks).
2. Protocol abuse and message broker overload (e.g., MQTT connection storms, CoAP amplification).
3. Edge gateway compromise and container breakout (e.g., privilege escalation within a Docker/containerd runtime at the edge).
4. Cloud control-plane attacks (e.g., update registry poisoning, Identity and Access Management (IAM) credential theft).

Crucially, an identical observable surface may represent both a reliability deficit and a severe security risk. For instance, a sudden spike in packet loss can obscure intentional network-level data poisoning, while a benign firmware defect triggering cyclic hardware watchdogs (reboots) may generate anomalous traffic patterns indistinguishable from a distributed denial-of-service (DDoS) payload [1]. Because architectural boundaries define how failures propagate [3, 5], practical resilience requires dynamic mathematical propagation modelling. A compromised node inflating connection requests can rapidly saturate an edge broker's memory pool, subsequently triggering unwarranted, costly cloud autoscaling — a systemic positive feedback loop that must be severed through explicit cross-tier evidence capture and rigid traffic containment rules.

Resilience patterns for identity, access control, and key lifecycle

In highly distributed architectures, identity and access control serve not merely as preventive perimeters, but as primary, highly effective recovery primitives. Upon the detection of anomalous behavior at the edge tier, rapid cryptographic key rotation and token revocation can halt fault propagation laterally across the network before a definitive, time-consuming root-cause analysis is completed. Standards strongly advocate against universal, hard-coded credentials to ensure that fleet-wide containment and granular, device-specific remediation remain computationally viable [4].

Managing the key lifecycle requires a delicate balance with embedded feasibility constraints. Resource-constrained devices should avoid frequent heavyweight asymmetric

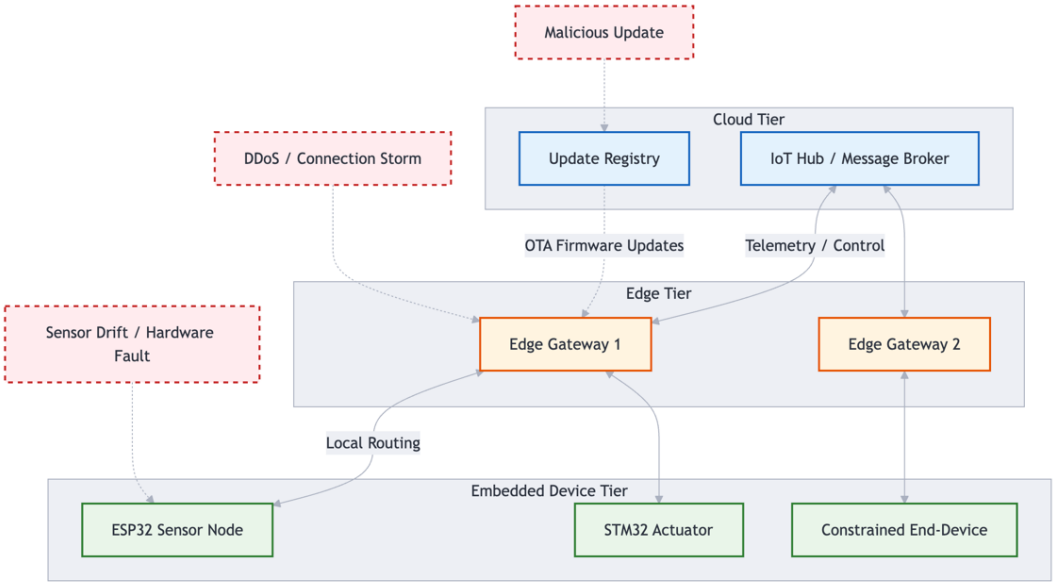


Fig. 1. Three-tier device–edge–cloud architecture illustrating the co-propagation of operational faults (e.g., sensor drift, battery depletion) and cyber-attacks (e.g., firmware modification, protocol injection) across the distributed continuum

cryptographic operations, while edge gateways can perform more complex identity negotiation and verification tasks. Microcontrollers are therefore better suited to minimal secure channels based on optimized symmetric cryptography, whereas event-driven signing, attestation, and re-enrollment are reserved for updates, anomalies, or scheduled audits. Cohort-based policies — grouping devices dynamically by operational criticality, topological proximity, and OTA update phase — further refine identity-driven containment, ensuring that critical recovery actions remain executable even under conditions of severe partial connectivity loss [4, 5, 6].

Update-centric resilience: TUF, SUIT, and Uptane patterns

Firmware update infrastructure constitutes a central point of systemic risk due to its unparalleled capacity to alter fleet-wide behavior simultaneously [12]. A robust design must operate under the pessimistic assumption that the cloud update repository itself may eventually be compromised by an Advanced Persistent Threat (APT). Frameworks such as TUF (The Update Framework) pioneered a compromise-resilience philosophy by demonstrating how cryptographically decoupled update authenticity, metadata signatures, and timestamp roles strictly limit the blast radius of exposed cryptographic keys [13].

In highly constrained environments, SUIT (Software Updates for Internet of Things) introduces compact binary metadata manifests (often CBOR-encoded) and explicit execution rules [14]. This permits microcontrollers to independently verify update integrity, validate dependencies, and assess local hardware conditions (such as ensuring sufficient battery level and memory availability before installation) before committing potentially destructive payloads to non-volatile flash memory. Uptane extends these principles directly into safety-critical operational environments (like automotive CPS), detailing procedures for surviving total repository infiltration [15]. Operationally, secure updates and fault tolerance exhibit deep interdependencies. A flawed but benign update can induce fleet-wide outages strictly analogous to a coordinated botnet attack. Consequently, OTA update rollouts must be orchestrated as tightly controlled, observable mathematical experiments tied directly to measurable telemetry stability indicators and cryptographically verified attestation logs [6, 13].

Observability engineering and dynamic evidence budgeting

Attaining sufficient observability on constrained devices necessitates a strict triage of telemetry signals due to the high energetic cost of continuous wireless transmission, which may dominate the total energy budget of battery-powered IoT nodes. Microcontrollers must therefore aggregate a minimal, high-value evidence set, typically comprising periodic heartbeats, firmware version hashes, energy state vectors, reboot counters, and statistical signal-quality features (variance, skewness). Edge nodes subsequently enrich this sparse data by appending queue latency metrics and executing cross-device temporal correlation, while the cloud tier trains models on long-horizon trends [2, 9]. This adaptive allocation mechanism is illustrated in **Fig. 2**, where evidence collection dynamically shifts between baseline telemetry and anomaly-driven diagnostics under strict energy constraints.

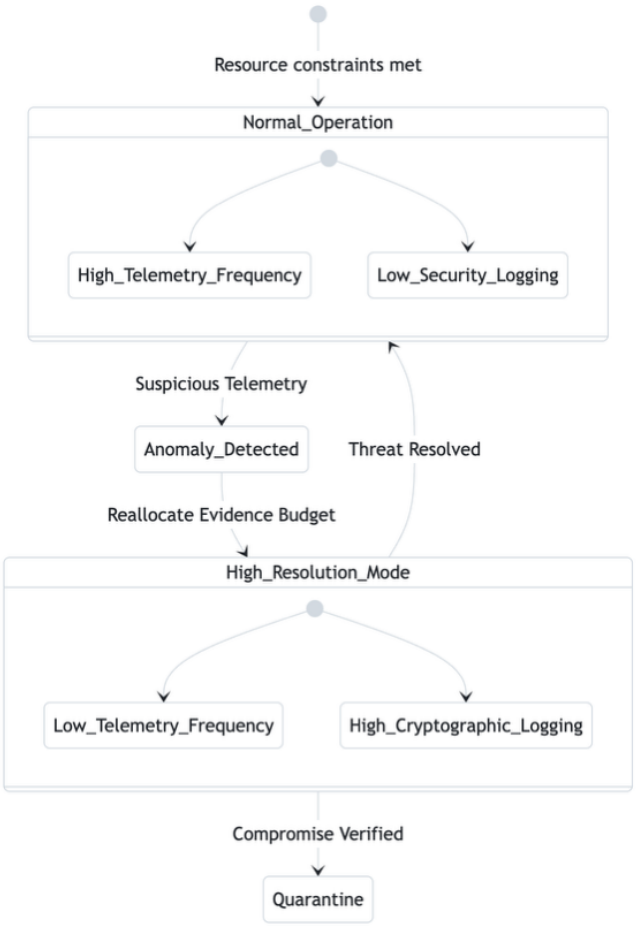


Fig. 2. Energy-aware evidence budgeting mechanism. The diagram demonstrates the dynamic reallocation of transmission frequencies (F_i) between baseline operational telemetry and high-resolution security evidence upon anomaly detection, strictly bounded by the total energy allowance B_{evid}

To prevent the observability apparatus from inducing secondary physical failures (such as rapid battery exhaustion via aggressive polling), telemetry collection must be dynamically adaptive, functioning strictly within a mathematically defined computational "evidence budget" (B_{evid}). Let c_i represent the energy cost per transmission of data type i, and f_i its respective frequency. The system must continuously satisfy the linear inequality:

$$\sum_{i \in \text{tasks}} c_i \cdot f_i \leq B_{evid}. \tag{8}$$

During normal, uninterrupted operation, security risks remain extremely low. However, following a major firmware update or the immediate detection of an anomaly, the edge orchestration policy dynamically reallocates weights to improve diagnostic accuracy, whilst strictly adhering to overall hardware constraints [9].

Detection and decision: evidence fusion and safety constraints

Algorithmic detection in cyber-physical platforms must be explicitly action-aware. A highly sensitive, low-threshold detection algorithm that triggers aggressive network isolation under conditions of mere statistical uncertainty can inadvertently degrade overall system resilience. This occurs by initiating unwarranted hardware reboots, network quarantines, and subsequent energy drain during reconnection handshakes. Therefore, detection efficacy must be quantified by its downstream impact on the system resilience curve rather than isolated, theoretical statistical metrics such as the probability of detection (P_d) and false positive rate (FPR) [7].

To minimize diagnostic ambiguity and reduce false positives, decision engines should implement rigorous evidence fusion. We propose a composite dynamic risk scoring formula $R(t)$ that synthesises real-time behavioural data-plane telemetry (denoted as $\text{Anomaly}(t)$) and historical cryptographic control-plane context (denoted as $\text{Trust}(t)$):

$$R(t) = \alpha \cdot \big(1 - \text{Trust}(t)\big) + \beta \cdot \text{Anomaly}(t) + \gamma \cdot \text{Degradation}(t), \tag{9}$$

where α, β, and γ are tuning coefficients reflecting environmental severity, and $\text{Degradation}(t)$ captures the hardware wear (e.g., sensor drift). Furthermore, automated recovery actions in classical IT systems (such as a hard reboot) may prove catastrophic in a CPS if they unexpectedly disable critical physical actuators. Consequently, self-healing algorithms must be bound by non-negotiable physical safety invariants, employing mathematical hysteresis loops and mandatory cooldown intervals to prevent damaging operational state oscillations [5, 7].

Edge orchestration and queuing latency implications

While edge orchestration platforms enable crucial localized processing during wide-area cloud partitions, their internal computational scheduling policies critically impact systemic resilience. Empirical evaluations of the KubeEdge framework indicate that routing traffic between disparately located edge nodes severely degrades data throughput and exponentially exacerbates response latency as inter-node physical distances increase [9].

We can model the edge message broker as an M/M/1 queuing system. If λ is the arrival rate of MQTT telemetry and μ is the service rate of the edge node, the average response time T is given by:

$$T = \frac{1}{\mu - \lambda}. \tag{10}$$

During an attack or a fault, the arrival rate λ may increase significantly. If λ approaches μ, the response time tends to infinity. These findings yield a vital resilience insight: pursuing "high availability" purely through topological distribution (sending data to a backup edge node) may prove illusory if the resulting network latency destabilizes local detection and physical control loops. Delayed broker responses will instigate client reconnection storms,

further amplify network load (λ) and mutate a localized software fault into a cascading, regional outage. Future research must, therefore, integrate joint orchestration-security policies, where pod placement and workload migration are determined not solely by CPU availability, but by the cryptographically attested trust state and the real-time risk scoring of the target node [6, 9].

Remote attestation: taxonomy and recovery integration

Remote attestation (RA) mechanisms offer varying trade-offs regarding measurement coverage, network scalability, and embedded resource expenditure. Rather than a singular monolithic protocol, RA typically operates across three distinct operational modes within a cloud-oriented platform [6]:

1. *Rollout verification:* Cryptographically validating the exact memory state immediately following an OTA firmware update.
2. *Anomaly response:* Triggering an on-demand, deep memory measurement when the edge gateway observes suspicious behavioral telemetry (i.e., when $R(t)$ exceeds a predefined threshold).
3. *Periodic auditing:* Executing low-frequency, randomized scheduled attestations to maintain a mathematical baseline of fleet assurance.

The general workflow of remote attestation in a constrained IoT environment is presented in **Fig. 3**.

RA yields maximum utility when directly coupled with the automated recovery loop. For instance, a network-quarantined node may only resume telemetry transmission post-attestation; similarly, rollback commands are only deemed successful once the restored

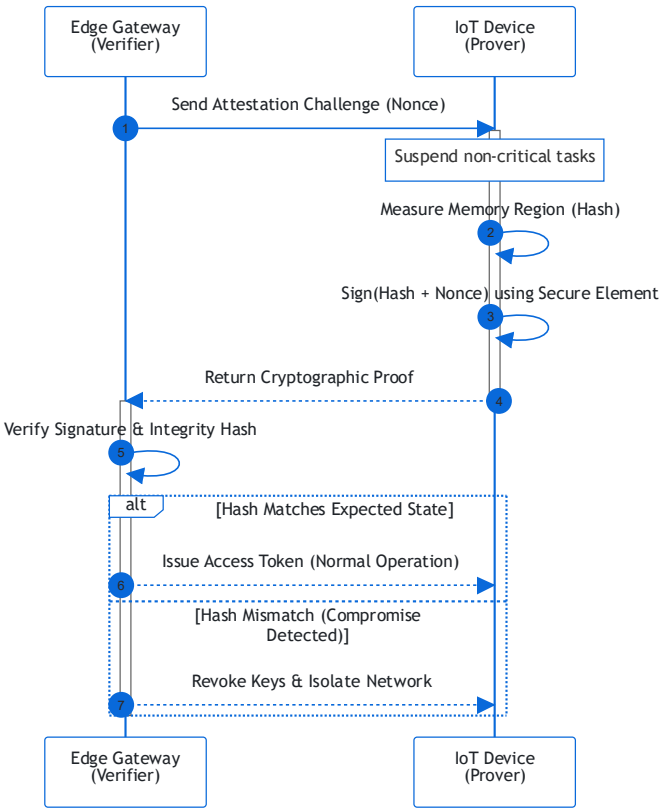


Fig. 3. Remote Attestation workflow. The sequence diagram illustrates the cryptographic challenge-response protocol between a resource-rich edge verifier and a constrained IoT prover

firmware hash is independently verified by the edge verifier. To conserve embedded energy, RA execution frequency must scale dynamically with the device's assigned mission criticality and its real-time anomaly score [6]. This adaptive attestation strategy enables a balance between assurance and resource efficiency in large-scale IoT deployments.

Supply-chain transparency and SBOM-guided recovery

Software Bill of Materials (SBOM) standards facilitate the precise, automated mapping of publicly disclosed vulnerabilities to specific active device fleets in the field. Although SBOM adoption across the software industry is accelerating, integration into continuous operational IoT environments remains a profound challenge due to device heterogeneity [11, 16].

Within the IoT context, SBOMs transition from static compliance documents to active, dynamic resilience tools when integrated directly into the control-plane decision matrix. Upon the disclosure of a critical Common Vulnerability and Exposure (CVE), an SBOM-aware engine can execute an automated graph-based impact assessment, immediately isolating affected cohorts and calculating the least disruptive remediation strategy (e.g., targeted memory patching versus temporary feature degradation). To function effectively at scale, these dependency graphs must be immutably linked to cryptographically signed device provenance records and live inventory registries [10, 11].

Digital twins and reproducible scenario suites

Digital twin methodologies provide indispensable, high-fidelity modelling environments for evaluating complex, non-linear resilience policies at scale without jeopardizing physical production systems [8]. In resilience engineering, digital twins are optimally utilized to execute vast, parameterized scenario suites, precisely calibrating detection thresholds (e.g., α, β, γ weights) and recovery delays before physical firmware deployment.

To ensure strict scientific rigor, these experimental suites must encompass a wide taxonomy of faults (e.g., precise network partitioning, memory starvation, sensor drift) alongside adversarial actions (e.g., credential hijacking, update repository poisoning). By structuring these scenarios through version-controlled injection scripts and explicit success criteria, researchers can achieve a hybrid evaluation model: leveraging the cloud simulation for vast statistical scaling, while utilizing the physical ESP32/STM32 testbed to capture empirical energy boundaries and network latency constraints, thereby continuously refining the fidelity of the digital twin [8].

Experimental protocol and resilience loss formalization

Establishing a rigorous mathematical and experimental protocol is foundational for validating the proposed dissertation methodology. The physical testbed topology, baseline firmware configurations, and multi-tier instrumentation strategies must be explicitly defined before execution.

A standard experimental trial should comprise distinct temporal phases: steady-state warm-up, precise fault/attack injection at time t_e, systemic response/detection at t_{det}, and verifiable recovery completed at t_{rec}. System performance is quantified not merely by timestamps, but by calculating the explicit "Resilience Loss" integral (R_{loss}). If Q_{target} is the targeted quality of service (e.g., 100% telemetry throughput) and $Q(t)$ is the actual measured quality over time, the total loss of the cyber-physical system is defined as:

$$R_{\text{loss}} = \int_{t_{\text{e}}}^{t_{\text{rec}}} \left[Q_{\text{target}} - Q(t)\right] dt. \tag{11}$$

Minimizing R_{loss} under the constraint $C_{\text{dev}} \leq C_{\text{max}}$ can be treated as a candidate optimization objective for future empirical studies. The temporal behavior of system

degradation and recovery, including the definition of resilience loss, is illustrated in **Fig. 4**. Given the inherent timing variability in wireless network stacks (e.g., Wi-Fi, LoRaWAN) and edge orchestrators, robust evaluation requires 10 to 30 repetitions per policy configuration. Results should be supported by 95% confidence intervals and non-parametric statistical testing (e.g., Mann-Whitney U test) to validate the practical significance of reviewed or derived self-healing policies against standard baseline configurations.

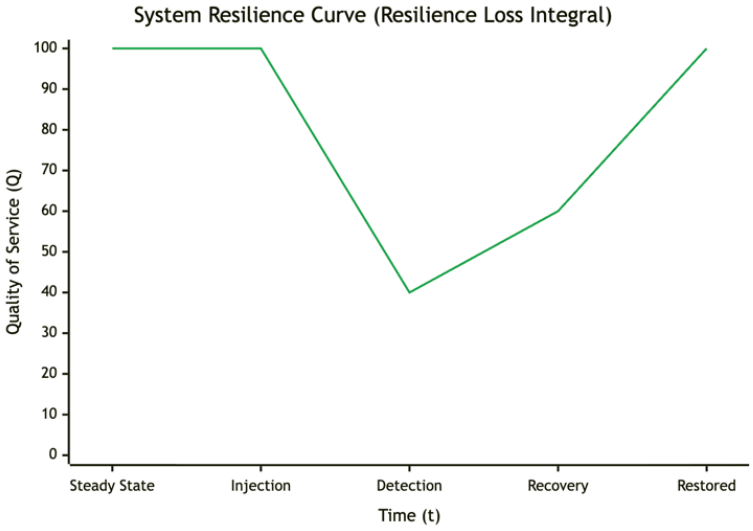


Fig. 4. System resilience curve showing service continuity degradation and recovery. The area above the dip represents the "Resilience Loss" integral ($\mathrm{R}_{\mathrm{loss}}$), captured between the point of fault injection (t_e) and full system recovery (t_{rec})

Summary of open problems and dissertation-aligned research agenda

Significant research gaps remain in the scientific literature regarding the development of unified resilience metrics that mathematically bridge cybersecurity (integrity, confidentiality) and classical dependability (reliability, safety). Furthermore, the scaling of risk-adaptive remote attestation and the execution of cryptographically verifiable firmware rollbacks under severe intermittent connectivity require significant architectural refinement.

Future research directions include the development of unified resilience metrics, reproducible fault and attack injection scenarios, risk-adaptive attestation strategies, and ESP32/STM32-KubeEdge testbed configurations for empirical validation. Such validation should determine whether the generalized model synthesized in this review can reduce resilience loss while respecting the computational, memory, communication, and energy constraints of the embedded domain.

CONCLUSIONS

This comprehensive review shows that fault tolerance in cloud-oriented IoT architectures is closely linked to cyber-resilience, since adversarial attacks and stochastic hardware faults can co-propagate across the distributed device, edge, and cloud continuum. Attaining practical operational resilience requires a shift from static, perimeter-based network defenses toward hierarchical evidence collection, update-centric lifecycle control, and automated self-healing mechanisms that respect established boundaries for embedded resource consumption (C_{max}).

We mathematically establish that remote attestation dramatically strengthens overall system recovery by supplying immutable, cryptographic integrity evidence to algorithmic decision engines [6]. Simultaneously, edge orchestration policies directly dictate the

queuing latency (T) – and therefore the dynamic stability – of physical containment loops [9]. Furthermore, integrating active SBOMs and strict provenance tracking facilitates rapid, fleet-scale vulnerability triaging, enabling significantly less disruptive remediation protocols [10, 11]. The subsequent phase of this ongoing dissertation research will focus directly on formalizing these risk-adaptive mathematical algorithms, minimizing the integral under energy constraints, and empirically validating the framework within a heavily constrained, multi-tier physical testbed.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [V.V., R.L.]; methodology, [V.V.]; validation, [V.V., R.L.]; writing – original draft preparation, [V.V.]; writing – review and editing, [V.V., R.L.]; visualization, [V.V.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog computing and IoT security: A survey and analysis. *Future Generation Computer Systems*, *78*, 680–698. https://doi.org/10.1016/j.future.2016.11.009

[2] Gkonis, P., Giannopoulos, A., Trakadas, P., et al. (2023). A Survey on IoT-Edge-Cloud Continuum Systems: Status, Challenges, Use Cases and Open Issues. *Future Internet*, *15*(12), 383. https://doi.org/10.3390/fi15120383

[3] ISO/IEC. (2024). ISO/IEC 30141:2024 Internet of Things (IoT) – Reference Architecture. https://cdn.standards.iteh.ai/samples/88800/e545dddaf5474251a0fb9af0e2109b9e/ISO-IEC-30141-2024.pdf

[4] ETSI. (2024). ETSI EN 303 645 V3.1.3: Cyber Security for Consumer Internet of Things: Baseline Requirements. https://www.etsi.org/deliver/etsi_en/303600_303699/303645/03.01.03_60/en_303645v030103p.pdf

[5] Ross, R. S., McEvilley, M., & Oren, J. C. (2018). Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems [including updates as of 1-03-2018]. https://doi.org/10.6028/NIST.SP.800-160v1

[6] Kuang, B., Fu, A., Susilo, W., Yu, S., & Gao, Y. (2022). A survey of remote attestation in Internet of Things: Attacks, countermeasures, and prospects. *Computers & Security*, *112*, 102498. https://doi.org/10.1016/j.cose.2021.102498

[7] Lezzi, M., Corallo, A., Lazol, M, Nimis, A. (2025). Measuring cyber resilience in industrial IoT: a systematic literature review. Management Review Quarterly. https://doi.org/10.1007/s11301-025-00495-8

[8] Epiphaniou, G., Hammoudeh, M., Yuan, G., et al. (2023). Digital twins in cyber effects modelling of IoT/CPS points of low resilience. *Simulation Modelling Practice and Theory*, *125*, 102744. https://doi.org/10.1016/j.simpat.2023.102744

[9] Kim, S.-H., & Kim, T. (2023). Local Scheduling in KubeEdge-Based Edge Computing Environment. *Sensors*, *23*(3), 1522. https://doi.org/10.3390/s23031522

[10] National Institute of Standards and Technology. (2022). Secure Software Development Framework (SSDF) Version 1.1 (NIST SP 800-218). https://doi.org/10.6028/NIST.SP.800-218
[11] Nocera, S., et al. (2025). On the adoption of software bill of materials in open-source software projects. *Journal of Systems and Software*, *230*, 112540. https://doi.org/10.1016/j.jss.2025.112540
[12] Catuogno, L., & Galdi, C. (2023). Secure Firmware Update: Challenges and Solutions. *Cryptography*, *7*(2), 30. https://doi.org/10.3390/cryptography7020030
[13] Samuel, J., Mathewson, N., Cappos, J., & Dingledine, R. (2010). Survivable key compromise in software update systems. *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 61–72. https://doi.org/10.1145/1866307.1866315
[14] Moran, B., Tschofenig, H., Brown, D., & Meriac, M. (2021). A Firmware Update Architecture for Internet of Things (RFC 9019). Internet Engineering Task Force. https://doi.org/10.17487/RFC9019
[15] IEEE-ISTO. (2022). IEEE-ISTO 6100.1.0.0 Uptane Standard for Design and Implementation of Secure Update Systems. https://uptane.org/papers/ieee-isto-6100.1.0.0.uptane-standard.pdf
[16] Stalnaker, T., Wintersgill, N., Chaparro, O., Di Penta, M., German, D. M., & Poshyvanyk, D. (2024). BOMs Away! Inside the Minds of Stakeholders: A Comprehensive Study of Bills of Materials for Software Systems. *Proceedings of the 46th ACM/IEEE International Conference on Software Engineering*, 1–13. https://doi.org/10.1145/3597503.3623347

МЕТОДИ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ТА КІБЕРСТІЙКОСТІ ХМАРНО-ОРІЄНТОВАНИХ ПЛАТФОРМ ІНТЕРНЕТУ РЕЧЕЙ В УМОВАХ ОБМЕЖЕНЬ ВБУДОВАНИХ ПРИСТРОЇВ

Володимир Войтович**, Роман Лис
Кафедра сенсорної та напівпровідникової електроніки,
Львівський національний університет імені Івана Франка,
вул. Генерала Тарнавського, 107, м. Львів, 79017, УКРАЇНА

АНОТАЦІЯ

Вступ. Хмарно-орієнтовані платформи Інтернету речей поєднують вбудовані сенсорні й виконавчі пристрої, периферійні вузли та хмарні сервіси. У таких архітектурах кіберзагрози й експлуатаційні відмови можуть поширюватися одночасно, знижуючи доступність, цілісність даних і фізичну безпеку. Забезпечення відмовостійкості та кіберстійкості за умов обмеженого енергоспоживання, пам'яті, обчислювальних ресурсів і зв'язку залишається актуальною науковою проблемою.

Матеріали та методи. Розглянуто архітектури типу «пристрій–периферійний рівень–хмарний рівень» для ресурсно-обмежених вбудованих систем. Узагальнено підходи системної інженерії, викладені в ISO/IEC 30141, NIST SP 800-160, ETSI EN 303 645, NISTIR 8259A, а також підходи до безпечного оновлення SUIT і TUF. Побудовано таксономію атак і відмов, систематизовано шаблони забезпечення стійкості для керування ідентичністю, моніторингу стану та відновлення працездатності з урахуванням деградації сенсорів. Теоретична частина охоплює метрики втрати стійкості, марковські моделі переходів станів і розподіл ресурсів для перевірки доказів довіри.

Результати. Показано, що практична кіберстійкість забезпечується поєднанням багаторівневого моніторингу, контролю життєвого циклу компонентів з урахуванням їхнього походження та механізмів самовідновлення, які діють за визначеними політиками й перевіряють інваріанти безпеки. Математична формалізація засвідчує, що підвищення доступності лише топологічним розподіленням ресурсів може порушувати контури стримування через затримки черг на периферійному рівні. Виявлено прогалини: нестачу уніфікованих метрик кібербезпеки й надійності, складність атестації пристроїв та обмежене виявлення загроз у мікроконтролерних вузлах.

Висновки. Відмовостійкість і кіберстійкість доцільно розглядати як адаптивний цикл «спостереження–виявлення–прийняття рішень–відновлення», оптимізований з урахуванням обмежень вбудованих систем. Робота формує основу для дослідження ризик-адаптивної віддаленої атестації та локально-орієнтованої оркестрації в середовищі ESP32/STM32–KubeEdge–AWS/Azure.

Ключові слова: кіберстійкість, відмовостійкість, хмарно-орієнтований Інтернет речей, периферійні обчислення, безпечні оновлення, самовідновлення.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
30 April, 2026	08 June, 2026	10 June, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 175–184
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 681.5, 004.93

INTEGRATION OF GRAPHENE-SENSOR-BASED ELECTRONIC NOSE TECHNOLOGY INTO MOBILE MICROCONTROLLER PLATFORM

Igor Olenych**, *Yaroslav Boyko*, *Bohdana Kukharska*, *Roman Yaremyk*, *Oleh Sinkevych*, *Mykhaylo Pavlyk* & *Ivan Karbovnyk
Faculty of Electronics and Computer Technologies,
Ivan Franko National University of Lviv,
50 Drahomanova Street, Lviv, 79005, Ukraine

Olenych, I., Boyko, Y., Kukharska, B., Yaremyk, R., Sinkevych, O., Pavlyk, R., & Karbovnyk, I. (2026). Integration of Graphene-Sensor-Based Electronic Nose Technology into Mobile Microcontroller Platform. *Electronics and Information Technologies*, 35, 175–184. https://doi.org/10.30970/eli.35.13

ABSTRACT

Background. The integration of sensors, in particular electronic nose technology, into mobile platforms is a fundamental element in the development of modern robotics and autonomous intelligent systems. This combination opens up new possibilities for automated air quality monitoring, detection of pollution sources, and the execution of rescue operations in hazardous conditions. An important aspect in this context is the use of artificial intelligence and machine learning algorithms to analyze sensor data and predict environmental changes. Consequently, particular attention was focused on implementing the microcontroller architecture of the intelligent sensor system.

Materials and methods. The developed mobile sensor platform architecture comprises a graphene FET-based sensor module, an impedance analyzer module, and a control module based on the ESP32 microcontroller. Normalized graphene channel resistance profiles and normalized gate resistance measured in atmospheres of acetone, ammonia, ethanol, and toluene vapor were used to develop a neural network-based machine learning model and test the intelligent sensor system.

Results. The developed model provided 90.9 % sensor data classification accuracy. The weighted average values for Precision, Recall, and F1-score are 0.94, 0.91, and 0.92, respectively, confirming the high quality of the classification and the model balance across the different classes of the target gases. The machine learning model deployed on the ESP32 microcontroller uses about 26% of flash memory and 6% of RAM for target gas recognition. The gas identification time, which does not exceed 1 ms, is small enough for the real-time intelligent sensor system.

Conclusions. The distributed architecture for the mobile sensor system based on the ESP32 microcontroller, combining control, sensor data measurement and acquisition, and data analysis functions, was proposed. It is demonstrated that the developed machine learning model provides high accuracy of target gas identification. The obtained results confirm the feasibility of implementing the proposed solutions on real-time platforms with limited computing resources.

Keywords: Graphene field-effect transistor, sensor platform, measuring system, microcontroller, machine learning model, gas recognition.

© 2026 Igor Olenych et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

Mobile robotic platforms represent a significant step toward operational efficiency and autonomy in various industries, the security sector, logistics, agriculture, and everyday life [1-3]. Central to this evolution is the integration of robotics, sensor systems, and artificial intelligence technologies, which are set to enhance robot adaptability to changing operating conditions and to ensure informed decision-making. Increasing demands for robot autonomy necessitate the use of a wide range of sensors to gather reliable information about the surrounding environment, enabling the localization and navigation of mobile platforms, as well as the real-time execution of specialized tasks. As robotic platforms perform increasingly complex tasks, their ability to understand their environment becomes crucial [4].

Sensor integration is a fundamental element of Industry 4.0, IoT, and unmanned technology development. In modern technological systems, mobile robots are part of a unified information space, exchanging data with other devices and information systems. An important aspect in this context is the implementation of multisensor data fusion methods, which enable the processing of information from various sources and the creation of a comprehensive picture of the environment [5, 6]. The use of artificial intelligence and machine learning algorithms makes it possible not only to analyze vast amounts of sensor data but also to predict changes in the state of objects or the environment. As a result, navigation reliability and operational robustness of mobile robots under conditions of uncertainty are increased.

The integration of electronic nose (e-nose) technology into mobile robotic platforms is of particular scientific and practical interest. The e-nose refers to an intelligent sensor system consisting of an array of gas sensors and pattern recognition algorithms for analyzing volatile organic compounds and gas mixtures [7-9]. Combining this technology with mobile robots opens up new possibilities for automated gas leak detection, air quality monitoring, pollution source localization, process diagnostics, and rescue operations in hazardous environments. At the same time, the platform mobility enables spatial mapping of gas concentrations and increases the informational value of the obtained data.

The combined analysis of data from different sensors, which form a unique multidimensional profile of the gaseous environment, is the basis of artificial olfaction. Recognition accuracy depends heavily on sensor stability, as drift in the characteristics of a large number of sensors creates a complex dependence of response on operating conditions [10]. The use of monosensor e-nose technology, on the one hand, simplifies the sensor system and reduces power consumption – factors important for mobile platforms and IoT – and on the other hand, increases the efficiency of gas identification over a long period of time. However, typical monosensor gas analyzers require additional thermal or ultrasonic influence on the sensing element [11, 12], which complicates the sensor system structure.

A promising solution to this problem could be the implementation of sensors based on the field-effect in graphene. The high sensitivity of graphene conductivity to local changes in the electric field induced by adsorbed gas molecules offers great potential for graphene-based materials in gas detection. Target gases are identified by analyzing the conductivity profiles of graphene field-effect transistors (FETs) measured in the studied environment using machine learning methods [13, 14]. Unlike thermal and ultrasonic effects on the sensor, measuring the dependence of the graphene FET channel resistance on the gate voltage is a relatively simple task using microprocessor technology [15]. This approach enables the integration of graphene sensors into microelectronic technology and the implementation of a portable gas analyzer based on a microcontroller-based measurement system architecture that meets the requirements of the IoT paradigm. Therefore, the work focused on investigating the feasibility of integrating

graphene FET-based e-nose technology into mobile microcontroller platforms and on developing tiny machine learning models capable of reliably identifying target gases.

MATERIALS AND METHODS

Sensor platform architecture

The developed mobile sensor platform architecture is a highly integrated system based on the ESP32 microcontroller (DD3), comprising a sensor module, an impedance analyzer module, and a control module (**Fig. 1**). The sensor module contains, as a sensing element, a FET based on a reduced graphene oxide (RGO) film deposited on the surface of porous silicon (PS) on a silicon substrate [14], as well as an analog interface. To obtain conductivity profiles of the graphene FET that are independent of the polarity of the voltage between the source and drain, the capability for AC measurements was provided.

The impedance analyzer module enables the software-controlled generation of a harmonic probe voltage V_{out} within a specified frequency range and with a set quantization step by the AD5933 impedance meter integrated circuit (DD2) from Analog Devices, as well as real-time impedance measurement using the Discrete Fourier

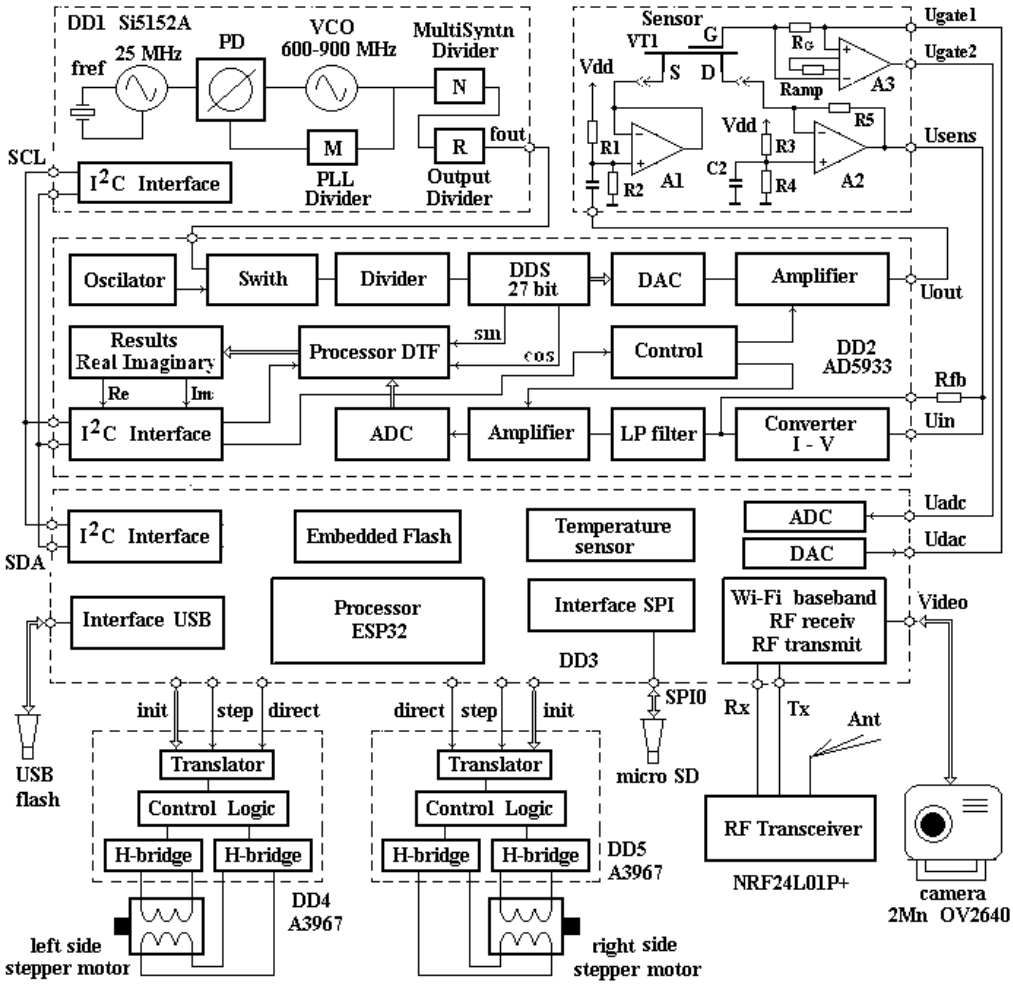


Fig. 1. Block diagram of a mobile sensor platform based on the ESP32 microcontroller

Transform [16]. According to the datasheet, the AD5933 can measure impedance in the 1–100 kHz range without any external components. The lower limit of the frequency range can be decreased using an external clock generator. The probe voltage induces a current through the graphene FET channel, the magnitude of which depends on both the gate voltage V_G and the gas type and concentration. The conversion of the measured information into a voltage V_{in} is implemented using an analog interface based on operational amplifiers A1 and A2. The real and imaginary parts of the impedance are determined by the discrete Fourier transform processor integrated into the AD5933 impedance meter chip for each gate voltage value within the programmed range at a specified frequency. The resulting internal resistance profiles of the graphene FET channel are stored in non-volatile memory.

Software control of the external clock generator Si5152A (DD1) and the frequency generator integrated into the DD2 chip is performed by an ESP32 microcontroller via the I^2C serial data bus. In addition to initializing the input registers and reading the output registers of the AD5933 impedance meter chip, the ESP32-based control module controls the FET gate voltage and the mobile platform motors, implements machine learning models, and transmits the video stream and control signals using an external RF transceiver NRF24L01P+. Installing the external transceiver enables motion control and video monitoring over greater distances (up to 100 m with the built-in antenna and 1 km in open space with an external antenna).

Drivers in the chips DD4 and DD5 of the A3967 type from Allegro MicroSystems Inc. are used to control the stepper motors of the mobile platform. Each microchip incorporates a ready-made electronic module for stepper motor control. A distinctive feature of the control scheme implementation is the use of an integrated translator that autonomously generates complex phase sequences for current in microstepping mode, while receiving only two logic pulse signals (step and direction) from the ESP32 microcontroller. This implementation of stepper motor control significantly reduces the load on the processor's computing resources. The translator A3967 automatically determines the current motion phase and switches the internal circuitry (H-bridges) to the next microstep. To control a standard bipolar stepper motor, two independent H-bridges in each driver are used, the current of which is regulated by a pulse-width modulated control circuit.

Data preparation

Experimental data measured in atmospheres of acetone, ammonia, ethanol, and toluene vapor within the 0–10 % concentration range were used to test the intelligent sensor system. The measured dataset consisted of the dependencies of the RGO–PS–Si-based FET resistance on the gate voltage for various concentrations of the target gases. **Fig. 2** illustrates the normalized resistance profiles of the RGO channel in the absence of target gases and in an atmosphere of acetone, ammonia, ethanol, and toluene.

Since the initial measurements form an irregular grid of gas concentration and gate-voltage values, their direct application in machine learning tasks is complicated by the lack of a consistent structure of input space. Therefore, multidimensional interpolation was applied to the experimental data to form a uniformly sampled feature space. In addition, the normalized resistance value of the sandwich structure – determined based on the gate leakage current for the corresponding target gas concentration – was added to the feature vector. The t-SNE dimensionality reduction method [17] was used to visualize the feature space (**Fig. 3**).

As a result, a unified labeled dataset containing consistent values for gas concentration, gate voltage, and the corresponding electrical characteristics of the sensor element was formed, enabling its direct application in classification algorithms. A dataset containing over 400 records was split into training and testing sets at an 80/20 ratio.

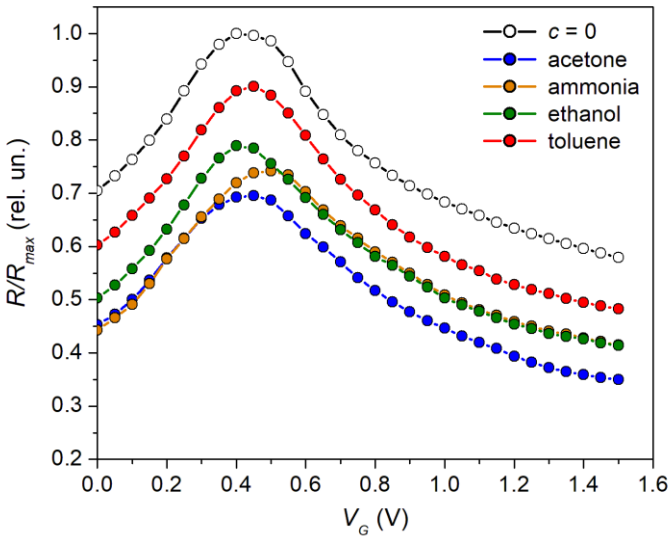


Fig. 2. The normalized resistance profiles of the RGO channel in the absence of target gases (c = 0) and in an atmosphere of acetone, ammonia, ethanol, and toluene at c = 6 % concentration

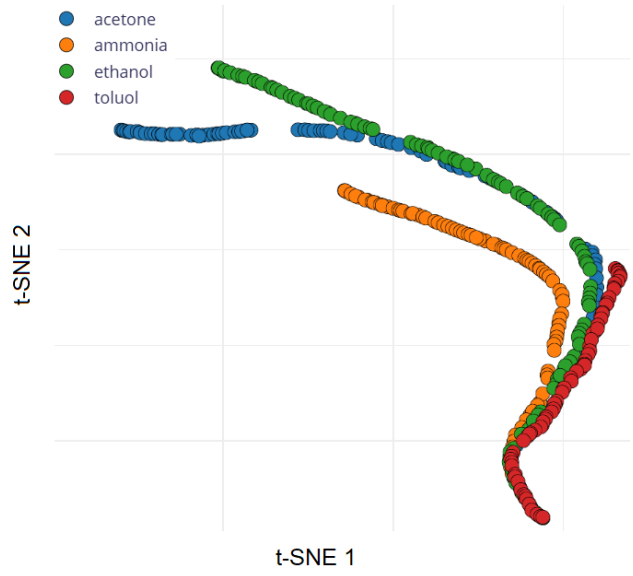


Fig. 3. Visualization of feature space

Model design

The classification model was based on a multilayer feedforward neural network. The input layer comprised 32 neurons, corresponding to the dimensionality of the feature vector. The input signal was then processed by two fully connected (dense) hidden layers with 20 and 10 neurons, respectively, using the ReLU activation function. The output layer contained four neurons, corresponding to the four target gas classes: acetone, ammonia, ethanol, and toluene. A Softmax activation function was applied to the output layer to convert the model outputs into a probability distribution over the

target classes. The learning rate was set to 0.0005. This value was selected empirically and, when used with the Adam optimizer during training, provided a stable learning process with a gradual improvement in classification accuracy.

RESULTS AND DISCUSSION

The trained neural network was tested on the validation set to evaluate the quality of the constructed model. The report on the effectiveness of the developed model is shown in **Fig. 4**. According to the confusion matrix, the "toluene" class demonstrated the highest recognition accuracy, with a correct classification rate of 100 %. For the "acetone", "ammonia", and "ethanol" classes, the proportion of correctly classified sensor data was 95.7, 88.2, and 83.3 %, respectively. Errors are related to the assignment of individual results measured in the atmosphere of other gases, probably at low concentrations, to the toluene class, since the accuracy of the classification depends on the concentration of the target gases [18].

Confusion matrix

	acetone	ammonia	ethanol	toluene
acetone	95.7 %	0	0	4.3 %
ammonia	0	88.2 %	0	11.8 %
ethanol	0	0	83.3%	16.7 %
toluene	0	0	0	100 %
F1 SCORE	0.98	0.94	0.91	0.79

Metrics

Metric	Value
Area under ROC Curve	0.99
Weighted average Precision	0.94
Weighted average Recall	0.91
Weighted average F1 score	0.92

Fig. 4. Confusion matrix and the report on the model effectiveness for multiclass classification

The developed model achieved a classification accuracy of 90.9 % with a loss function value of 0.42. The obtained results indicate the ability of the model to correctly recognize the studied gases using data not included in the training set. The area under the ROC curve is 0.99, which serves as further evidence of the high class-discrimination capability. The weighted average values for Precision, Recall, and F1-score are 0.94, 0.91, and 0.92, respectively, confirming the high quality of the classification and the model balance across the different classes of the target gases. Dimensionality reduction was used to visualize the classification results. In **Fig. 5**, which illustrates the resulting diagram, correctly classified sensor data are marked with various shades of green, while incorrectly classified data are marked with shades of red. The majority of points belong to correctly classified data, confirming a high level of classification accuracy based on the derived sensory features.

The trained model was deployed on an ESP32 microcontroller to implement the developed sensor data analysis system. To verify the performance of the qualification model on the microcontroller, four test feature vectors corresponding to the target gases:

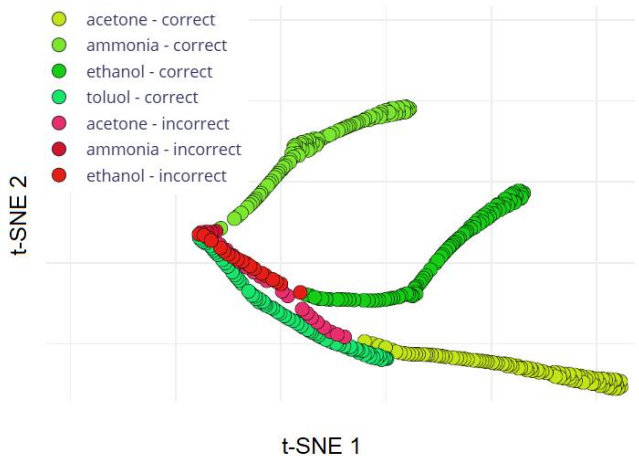


Fig. 5. Confusion matrix and the report on the model effectiveness for multiclass classification

acetone, ammonia, ethanol, and toluene were used. For these vectors, the model determined the probability of the sensor data belonging to each of the four classes and generated the classification results, which are presented in **Table 1**.

Analysis of the results in **Table 1** showed that the model deployed on the microcontroller correctly identified all target gases. The model identified ethanol with the highest confidence (almost 97.3 % probability). The probability of correctly identifying acetone and ammonia exceeded 80 %, indicating a sufficiently high classification quality. Even for the case of the lowest probability of toluene identification (54.3 %), the developed model successfully performed the classification. In addition, the gas identification time of the intelligent sensor system was evaluated. The time required to obtain a conclusion regarding the correspondence of each of the vectors to the target gases did not exceed 1 ms. Such performance confirms the feasibility of using the developed model for real-time sensor data analysis by the ESP32 microcontroller.

Table 1. **The probability of the test data belonging to "acetone", "ammonia", "ethanol", and "toluene" classes**

Actual	Predicted			
	acetone	ammonia	ethanol	toluene
acetone	0.84375	0.02734	0.01953	0.16406
ammonia	0.01172	0.81641	0.00781	0.07422
ethanol	0.08203	0.12500	0.97266	0.22266
toluene	0.05859	0.02734	0.00000	0.54297

Finally, the use of ESP32 microcontroller hardware resources was analyzed. The program code size, including the machine learning model, occupied 26 % of the available flash memory. And only 6 % of the microcontroller RAM was used to store global variables. Thus, the gas recognition model implementation leaves a significant amount of both flash memory and RAM free, which not only does not interfere with control of the mobile platform but also allows for the expansion of its functionality without the need to upgrade the hardware.

CONCLUSION

The developed intelligent sensor system is proposed as a universal mobile platform based on the ESP32 microcontroller, combining control, sensor data measurement and acquisition, and data analysis functions. A distinctive feature of the hardware implementation of the developed sensor system is the ability to configure the measurement signal path and the measurement algorithm via software, taking into account the specific characteristics of the sensor response of the FETs based on the RGO–PS–Si structure. Local processing and analysis of the measured sensor data are performed directly by the microcontroller-based mobile platform. Sensor data classification is performed by the trained neural network deployed on an ESP32. The developed machine learning model demonstrates 90.9 % accuracy in recognizing gases (acetone, ammonia, ethanol, and toluene) and uses about 26 % of flash memory and 6 % of RAM for classification. The gas identification time does not exceed 1 ms. The obtained results confirm the feasibility of implementing the proposed solutions in real time on resource-constrained devices, particularly on mobile microcontroller platforms. The openness of the architecture to expansion at both hardware and software levels enables optimal configuration of the measurement parameters and enhances the functionality of the mobile platform.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by the Ministry of Education and Science of Ukraine [0124U000982].

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [I.O., Y.B., I.K.]; methodology, [I.O., Y.B., B.K., O.S.]; investigation, [I.O., B.K., R.Y., M.P.]; writing – original draft preparation, [I.O., R.Y., O.S.]; writing – review and editing, [I.O., O.S.]; visualization, [I.O., B.K., R.Y.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Boleraczki, M. & Gyurika, I.G. (2025). Industrial mobile robot-based manufacturing system modeling potential. *Data-Centric Engineering*, 6, e46. https://doi.org/10.1017/dce.2025.10020

[2] Pavel, M.-D., Stamatescu, G., Chodnicki, M., & Amza, C. G. (2026). LLM-Enhanced Control of a Mobile Robotic Platform for Smart Industry. *Applied Sciences*, 16(4), 1680. https://doi.org/10.3390/app16041680

[3] Al-Tawil, B., Candemir, A., Jung, M., & Al-Hamadi, A. (2025). Mobile Robot Navigation with Enhanced 2D Mapping and Multi-Sensor Fusion. *Sensors*, 25(8), 2408. https://doi.org/10.3390/s25082408

[4] Placed, J.A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., & Castellanos, J.A. (2023). A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Trans. Robot.*, 39, 1686–1705. https://doi.org/10.1109/TRO.2023.3248510

[5] Fan, Z., Zhang, L., Wang, X., Shen, Y., & Deng, F. (2025). LiDAR, IMU, and camera fusion for simultaneous localization and mapping: a systematic review. *Artif. Intell.*

Rev., 58, 174.
https://doi.org/10.1007/s10462-025-11187-w
[6] Usinskis, V., Nowicki, M., Dzedzickis, A., & Bucinskas, V. (2025). Sensor-Fusion Based Navigation for Autonomous Mobile Robot. *Sensors*, 25(4), 1248. https://doi.org/10.3390/s25041248
[7] Zhai, Z., Liu, Y., Li, C., Wang, D., & Wu, H. (2024). Electronic Noses: From Gas-Sensitive Components and Practical Applications to Data Processing. *Sensors*, 24(15), 4806. https://doi.org/10.3390/s24154806
[8] Ma, D., Gao, J., Zhang, Z., & Zhao, H. (2021). Gas recognition method based on the deep learning model of sensor array response map. *Sensor Actuat. B Chem*., 330, 129349. https://doi.org/10.1016/j.snb.2020.129349
[9] Zhang, W., Tang, W., Wan, Z., & Fan, Z. (2026). Advanced electronic noses for future robotic olfaction. *npj Robot*, 4, 11. https://doi.org/10.1038/s44182-025-00071-y
[10] Liu, H., Chu, R.Z., & Tang, Z.A. (2015). Metal oxide gas sensor drift compensation using a two-dimensional classifier ensemble. *Sensors*, 15, 10180. https://doi.org/10.3390/s150510180
[11] Huang, J.R., Gu, C.P., Meng, F.L., Li, M.Q., & Liu, J.H. (2007). Detection of volatile organic compounds by using a single temperature-modulated SnO2 gas sensor and artificial neural network. *Smart Mater. Struct*. 16, 701. https://doi.org/10.1088/0964-1726/16/3/017
[12] Su, S. & Hu, J. (2019). Gas identification by a single metal-oxide-semiconductor sensor assisted by ultrasound, *ACS Sens*. 4, 2491. https://doi.org/10.1021/acssensors.9b01113
[13] Hayasaka, T., Lin, A., Copa, V. C., Lopez Jr., L.P., Loberternos, R.A., Ballesteros, L.I.M., Kubota, Y., Liu, Y., Salvador, A.A., & Lin, L. (2020). An electronic nose using a single graphene FET and machine learning for water, methanol, and ethanol. *Microsyst. Nanoeng*. 6, 50. https://doi.org/10.1038/s41378-020-0161-3
[14] Olenych, I. (2025). Field-effect transistor based on reduced graphene oxide – porous silicon hybrid structure for application in e-nose technology. *Mol. Cryst. Liq. Cryst*., 769, 1527–1537. https://doi.org/10.1080/15421406.2025.2540098
[15] Olenych, I., Yaremyk, R., & Parandiy, P. (2025). Microprocessor system for measuring sensor characteristics of gas-sensitive field-effect transistors based on porous silicon and reduced graphene oxide film. *Computer-Integrated Technologies: Education, Science, Production*, 61, 171–177 (in Ukrainian). https://doi.org/10.36910/6775-2524-0560-2025-61-24
[16] Simic, M. (2015). Complex Impedance Measurement System for the Frequency Range from 5 kHz to 100 kHz. *Key Engineering Materials*. 644, 133–136. https://doi.org/10.4028/www.scientific.net/KEM.644.133
[17] van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*. 9, 2579-2605. http://jmlr.org/papers/v9/vandermaaten08a.html
[18] Olenych, I. & Osadchuk, O. (2024). Application of Machine Learning Algorithms in Electronic Nose Technology. *Electronics and Information Technologies*. 25, 56–66 (in Ukrainian). https://doi.org/10.30970/eli.25.5

ІНТЕГРАЦІЯ ТЕХНОЛОГІЇ ЕЛЕКТРОННОГО НОСУ НА ОСНОВІ ГРАФЕНОВОГО СЕНСОРА У МОБІЛЬНУ МІКРОКОНТРОЛЕРНУ ПЛАТФОРМУ

Ігор Оленич*, Ярослав Бойко, Богдана Кухарська, Роман Яремик, Олег Сінькевич, Михайло Павлик, Іван Карбовник
Факультет електроніки та комп'ютерних технологій
Львівський національний університет імені Івана Франка
вул. Драгоманова 50, м. Львів, 79005, Україна

АНОТАЦІЯ

Вступ. Інтеграція сенсорів, зокрема технології електронного носа, у мобільні платформи є фундаментальним елементом розвитку сучасної робототехніки та автономних інтелектуальних систем. Таке поєднання відкриває нові можливості для автоматизованого моніторингу якості повітря, виявлення джерел забруднення і проведення рятувальних операцій у небезпечних умовах. Важливим аспектом у цьому контексті є використання штучного інтелекту та алгоритмів машинного навчання для аналізу сенсорних даних і прогнозування зміни навколишнього середовища. Відтак, особлива увага була зосереджена на реалізації мікроконтролерної архітектури інтелектуальної сенсорної системи.

Матеріали та методи. Розроблена архітектура мобільної сенсорної платформи містить сенсорний модуль на основі графенового польового транзистора, модуль аналізатора імпедансу та модуль керування на базі мікроконтролера ESP32. Нормовані профілі опору графенового каналу та нормований опір затвора, виміряні в атмосферах парів ацетону, аміаку, етанолу і толуолу, були використані для розробки моделі машинного навчання на основі нейронної мережі та тестування інтелектуальної сенсорної системи.

Результати. Розроблена модель забезпечила точність класифікації сенсорних даних 90,9 %. Середньозважені значення для влучності (англ. precision), повноти (англ. recall) і F1-міри (англ. F1-score) становлять 0,94, 0,91 і 0,92 відповідно, що підтверджує високу якість класифікації та збалансованість моделі для різних класів цільових газів. Модель машинного навчання, розгорнута на мікроконтролері ESP32, використовує близько 26 % флеш-пам'яті та 6 % оперативної пам'яті для розпізнавання цільового газу. Час ідентифікації газу, який не перевищує 1 мс, достатньо малий для функціонування інтелектуальної сенсорної системи в реальному масштабі часу.

Висновки. Запропоновано розподілену архітектуру мобільної сенсорної системи на основі мікроконтролера ESP32, що поєднує в собі функції керування, вимірювання та збору даних датчиків та аналізу даних. Продемонстровано, що розроблена модель машинного навчання забезпечує високу точність ідентифікації цільових газів. Отримані результати підтверджують можливість реалізації запропонованих рішень на платформах реального часу з обмеженими обчислювальними ресурсами.

Ключові слова: Графеновий польовий транзистор, сенсорна платформа, вимірювальна система, мікроконтролер, модель машинного навчання, розпізнавання газів.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
10 September, 2026	18 September , 2026	21 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 185–204
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.35:004.5:004.62:681.5

OPTIMIZATION OF AN AUTOMATED ARDUINO-BASED SYSTEM FOR CHARACTERIZATION OF DISCRETE OPTOELECTRONIC DEVICES

Mykhailo Rusyniak , ***Yuriy Korchak**** , ***Bohdan Kuzhel***
Department of Optoelectronics and Information Technology
Ivan Franko National University of Lviv
107 Gen. Tarnavskoho Street, Lviv, 79017, UKRAINE

Rusyniak, M., Korchak, Yu., Kuzhel, B. (2026). Optimization of an Automated Arduino-based System for Characterization of Discrete Optoelectronic Devices. *Electronics and Information Technologies*, 35, 185–204. https://doi.org/10.30970/eli.35.14

ABSTRACT

Background. Qualitative investigation of the electrical and optical characteristics of optoelectronic components is an essential part of designing, developing, and testing optoelectronic systems. For this purpose, an automated system for measuring the characteristics of optoelectronic devices was developed based on an Arduino UNO microcontroller, featuring programmable voltage control via digital potentiometers and data transfer to a computer for visualization. Additionally, the system was optimized to enhance its performance.

Materials and Methods. A fully functional automated system for measuring the voltage–current and optical characteristics of optoelectronic devices has been developed and implemented based on the Arduino UNO microcontroller. The hardware part of the system is built using MCP4131 digital potentiometers and an integrated GY-302 light sensor. The desktop application was developed in Python, utilizing the Tkinter, pySerial, Matplotlib, and Threading libraries. Adaptive system optimization was carried out using the finite difference method to compute the corresponding local derivatives.

Results and Discussion. Experimental investigation of the optimized system's performance demonstrated a significant increase in efficiency while maintaining high-quality characteristics. The developed adaptive optimization system showed a stable acceleration of 2.4-4.3 times, depending on the type of component. During the experiments, no cases of missing critical regions or significant distortion of characteristic curves were observed. This confirms the reliability of the developed zone-classification system and the correctness of the selected adaptive algorithm parameters.

Conclusion. The results demonstrate the feasibility of creating an efficient and accessible system for studying the characteristics of optoelectronic devices based on modern microcontroller platforms. The developed adaptive optimization system proved effective under real operating conditions, making it a valuable tool for both educational and practical applications in the research and development of optoelectronic devices.

Keywords: current–voltage characteristic, finite difference method, Arduino microcontroller, LED, photodiode, digital potentiometer

INTRODUCTION

Over the past decades, optoelectronics has evolved into one of the most promising branches of modern electronics. It encompasses a wide range of applications in lighting engineering, displays, solar energy systems, fiber-optic communication channels, sensor technologies, and the like. This popularity is driven by the ability of optoelectronic devices

© 2026 Mykhailo Rusyniak et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

to efficiently convert light and electrical energy into one another, as well as by the high information capacity and energy density of the optical channel, which opens up wide opportunities for technical solutions across various fields [1, 2].

Optoelectronic systems often include discrete emitting and photodetector devices. The electrical and optical characteristics of these elements significantly affect the efficiency and stability of the entire electronic device. Therefore, precise investigation of their parameters is a necessary component during the design, development, and testing of optoelectronic systems. However, measuring such characteristics by traditional methods typically requires fairly expensive laboratory equipment. In addition, the manual data collection process is labor‑intensive, inefficient, and not always accurate. Even with process automation, traditional approaches with a fixed scanning step lead to an excessive number of measurements in predictable regions of the characteristic, making the process time‑inefficient.

Given the accessibility, ease of use, and functionality of modern microcontrollers, an automated experimental setup has been developed for investigating the characteristics of discrete optoelectronic components using the Arduino UNO [3, 4]. Arduino enables the creation of simple yet efficient data acquisition and processing systems, making it an ideal tool for both educational purposes and basic laboratory research. The system provides programmable control of voltage and current applied to the device under test via digital potentiometers, allows measurement of electrical parameters using the microcontroller's built-in capabilities, and facilitates data transfer to a computer for visualization and analysis. The approximate cost of this system (excluding the PC) is up to $50.

To address the problem of redundant measurements in linear and predictable regions of a device's characteristic under a fixed voltage step; an adaptive measurement optimization system was developed. This system is based on numerical analysis techniques and dynamic classification of characteristic regions. It utilizes the finite difference method to calculate local derivatives [5], which are then used to classify the current segment of the characteristic and dynamically adjust the scanning step size.

MATERIALS AND METHODS

The development of a compact, software-integrated system for investigating the electrical and optical characteristics of optoelectronic components in laboratory conditions involves both hardware and software design [6].

The hardware design includes the electrical circuit connecting optoelectronic elements to the Arduino Uno, ensuring device safety and measurement reliability. The cloud-based platform EasyEDA was used for circuit and printed circuit board (PCB) design, offering schematic editing, PCB layout, and component simulation. EasyEDA facilitates component placement, connection verification, signal and power trace design, and ergonomic layout assessment, while enabling seamless export to formats suitable for PCB fabrication.

The prototype also incorporates a physical structure with optimally arranged components and adjustable lighting to enhance measurement accuracy and minimize external influences. Two controllable voltage sources allow applying both positive and negative voltages from −7 to +7 V, while current is measured via Arduino's analog-to-digital converter (ADC) using a shunt resistor. Illumination monitoring is provided by an integrated GY-302 lux meter module.

Arduino Uno serves as the central controller, acquiring sensor data, controlling voltage, and transmitting measurements to a computer via a serial interface. A Python-based program automates experiments, performs data processing and calibration, and visualizes results. It also coordinates voltage settings, lighting control, and measurement sequencing, enabling fully automated experimental workflows.

This integrated hardware-software approach provides precise control and analysis of electrical and optical parameters, offering a versatile platform for educational, laboratory, and research applications. The system allows rapid reconfiguration for different experimental scenarios, combining measurement accuracy with user-friendly operation and automation.

The software component of the prototype is implemented as a single Python file that integrates all functional elements of the system. Despite this unified code structure, the program is logically divided into two main blocks: the graphical user interface (GUI) and the hardware interaction module. This approach facilitates organized code, improves readability, and separates logic from data presentation [7].

The GUI block is responsible for displaying control elements and experimental results. It includes buttons for connecting the Arduino, calibrating the system, selecting the optoelectronic device, and starting the automated measurement process. Additionally, the GUI presents results in the form of graphs and tables. The visual interface not only provides informative output but also enables interactivity: users can adjust experiment parameters in real time, monitor measurements, and, if necessary, restart individual stages.

The second block manages hardware interaction, specifically with the Arduino microcontroller. This module implements communication via the serial port, allowing commands to control voltage sources, acquire data from ADCs and sensors, and perform prototype calibration. All operations are automated: once the experiment begins, the Python program controls the voltage applied to the device under test, reads measured current and illumination values, and transfers the data for processing and visualization.

To enhance GUI responsiveness and efficiency, multithreading is employed: one thread handles data processing and Arduino communication, while a second thread updates the interface and plots graphs in real time. This approach prevents program lag and ensures immediate visibility of experimental results, even during computations.

Several key Python libraries are utilized to handle data acquisition, prototype control, and result visualization (Tkinter, pySerial, Matplotlib, Threading), each performing a specific function and simplifying software implementation.

Tkinter is the standard Python library for creating graphical user interfaces (GUI) [8], allowing the development of windowed applications with buttons, input fields, sliders, and graphical widgets. Tkinter facilitates rapid integration of control elements without relying on additional graphical frameworks.

pySerial is a library for working with serial interfaces (COM ports) [9], enabling data exchange between the computer and the Arduino microcontroller via USB.

Matplotlib is a library for plotting and visualizing data [10], used for visually presenting experimental results. It supports line, scatter, histogram, and other types of plots, making data analysis more intuitive [11, 12].

Threading is a Python library for implementing multithreading [13], allowing simultaneous execution of multiple tasks without blocking the user interface. In this system, threading enables parallel data acquisition from the Arduino, signal processing, and real-time graph updates.

The combination of these libraries provides a fully functional software component, covering experiment control, data acquisition, processing, and visualization. Using Python and its libraries ensures system flexibility, scalability, and suitability for further expansion or integration with other sensor systems. By combining modularity, automation, and an intuitive GUI, the software component becomes an effective tool for educational and scientific experiments with optoelectronic devices.

RESULTS AND DISCUSSION

Development and testing of a desktop application

Using the computer-aided design (CAD) tool EasyEDA [14], the electrical schematic of the experimental prototype was created (**Fig. 1**). The circuit is based on two adjustable voltage regulators, LD1117-ADJ. The input voltage is supplied from the *Vin* pin of the Arduino Uno, while the output voltage is controlled by voltage dividers consisting of resistor R1 (1 kΩ) and a digitally controlled potentiometer P1 (MCP4131-502 [15]) for voltage regulator U1, and resistor R2 (1 kΩ) with a digitally controlled potentiometer P2 (MCP4131-502 [15]) for voltage regulator U2. The output voltage is determined by the following relationship:

$$V_{out} = V_{ref} \cdot \left(1 + \frac{R_{2a}}{R_{1a}}\right), \quad (1)$$

where V_{ref} = 1.25 V, R_{2a} represents the resistance of the digitally controlled potentiometer (P1 or P2 in Circuit Diagram), and R_{1a} corresponds to the resistance of the fixed resistor R1 or R2, depending on the regulator.

Voltage adjustment is performed using digitally controlled potentiometers MCP-4131-502. Their maximum resistance is 5 kΩ (±20%), and the register controlling the resistance has a 7-bit resolution, providing 128 possible resistance values. The wiper resistance, approximately 75 Ω, must also be taken into account. In practice, with 9 V applied to the *Vin* pin, output voltages of 2.15-7 V were obtained. This voltage range from both sides allows applying up to 7 V in either forward or reverse direction.

To measure high currents, the circuit includes resistor R3, which also acts as a current limiter. For low-current measurements, a digitally controlled potentiometer MCP4131-503 (50 kΩ) and a 10 kΩ resistor were added to the circuit, enabling measurement of currents on the order of microamperes for investigating the current–voltage characteristics (*I-V* characteristic) of the photodiode. Illumination measurements are performed using a GY-302 lux meter.

Based on this electrical schematic, a custom PCB was manufactured. The board's form factor replicates the dimensions of the Arduino Uno, serving as a shield (**Fig. 2a**).

To ensure accurate measurements, the system requires light shielding to eliminate any influence of external illumination. Additionally, user convenience and protection of system components from physical impact are necessary. For this purpose, a case was

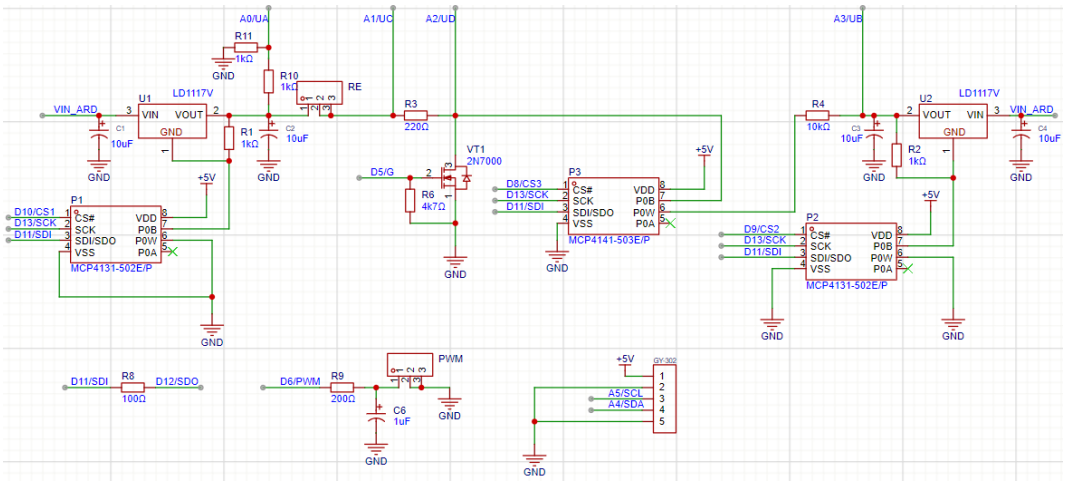
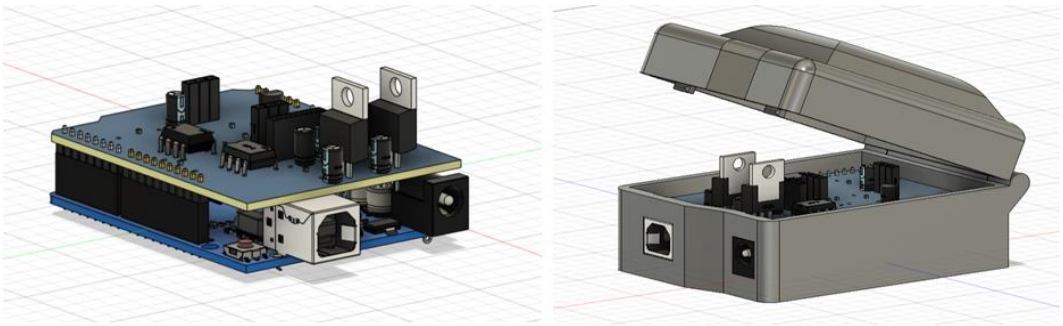


Fig. 1. Circuit Diagram of the Prototype



a) b)

Fig. 2. Experimental prototype: board integrated with Arduino Uno (a), protective enclosure (b)

designed, shown in **Fig. 2b**, optimized for 3D printing and taking into account all system requirements.

As mentioned above, the cross-platform library Tkinter was chosen for implementing the GUI. Tkinter is a standard Python library that does not require additional installation and is well-suited for developing desktop applications due to its sufficient set of widgets for creating a functional interface.

The application interface, shown in **Fig. 3**, is built on a modular principle and consists of three main sections: a control panel and two graph display areas. This structure allows users to conveniently manage the measurement process, monitor results in real time, and receive information about the system status.

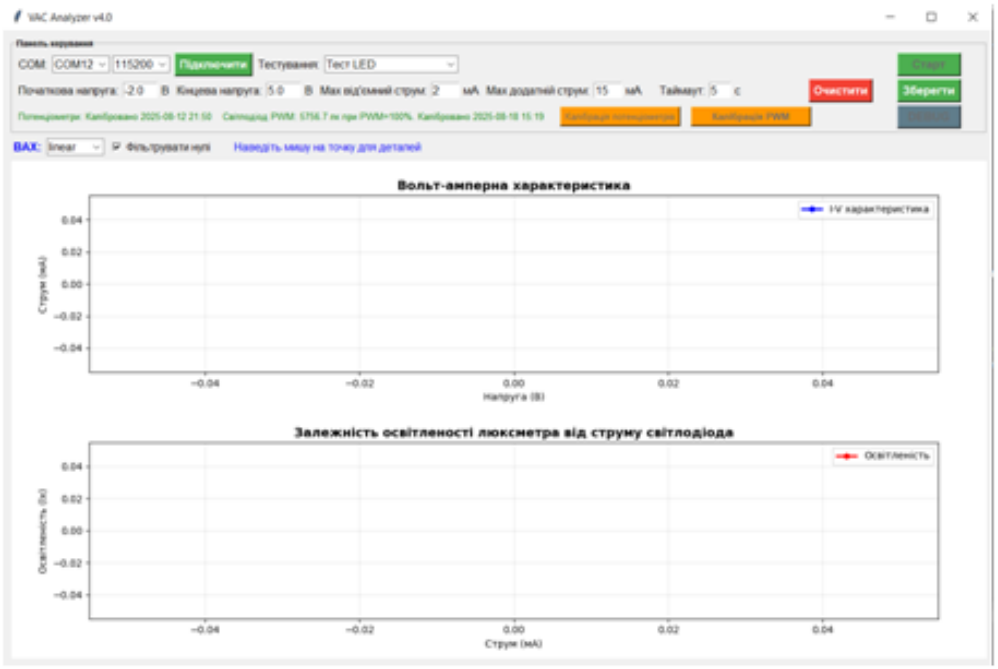
The control panel is located at the top of the application window and is organized into three horizontal rows of control elements. The first row contains controls for connecting to the microcontroller and selecting the type of testing. The application supports two operation modes: light-emitting diode (LED) testing (**Fig. 3a**) and photodetector testing (**Fig. 3b**). When the photodetector testing mode is selected, an additional field appears for setting the pulse-width modulation (PWM) value of the illuminating LED (see **Fig. 3b**). The second row of the control panel contains parameters for configuring the measurement process, while the third row allows system calibration. This row displays the status of potentiometer and LED PWM calibration.

The graph display area occupies the main part of the application window and is implemented using the Matplotlib library. Integration of Matplotlib with Tkinter is achieved via the *FigureCanvasTkAgg* module, which enables embedding graphs directly into the application window. The interface includes two vertically stacked graphs: the upper graph displays the *I-V* characteristic of the device under test, while the lower graph shows the dependence of illumination on current for LEDs. When working with photodetectors, the lower graph is automatically hidden, as it provides no relevant information for this type of device. Instead, a text field appears at the top, displaying the current illumination value in lux.

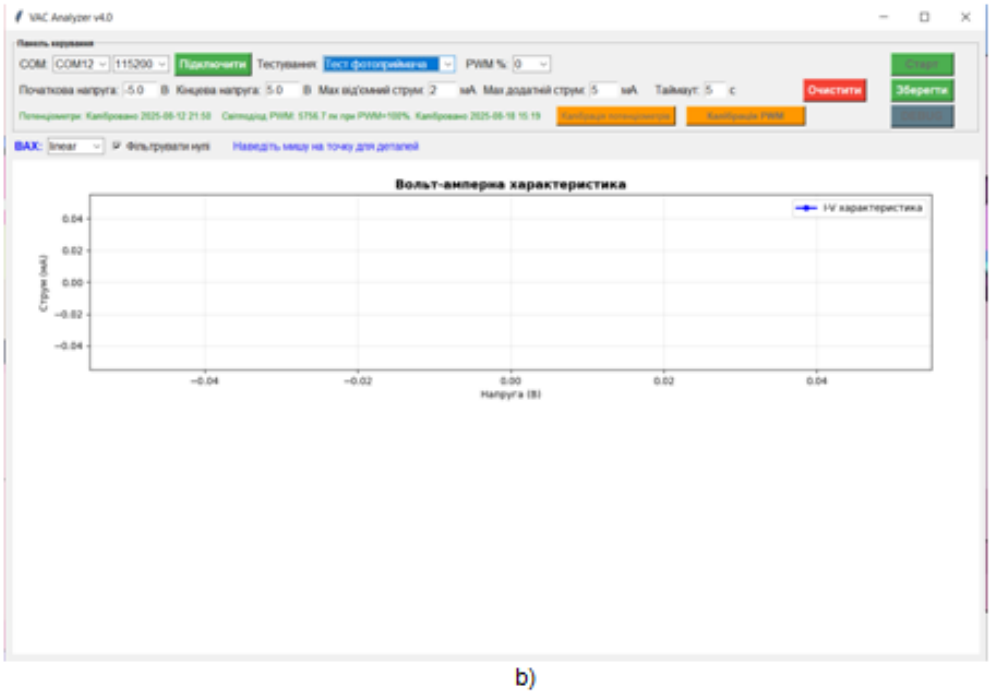
Above the graphs, an additional settings panel is located, allowing users to change the display mode. The user can switch between linear and logarithmic scales. A zero-value filtering option is also provided, which is enabled by default. This prevents points with zero current from being displayed on the graph, improving the readability of the characteristics.

A key feature of the graphs is their interactivity. When the cursor hovers over any point on the graph, a tooltip automatically appears, providing detailed information about that point.

During the interface development, special attention was paid to the logic of control element states. Many buttons and input fields are automatically enabled or disabled depending on the system state. For example, the "Start" button is only available when a connection to the microcontroller is established, and the system is properly calibrated. During measurements, input fields for test parameters are locked to prevent accidental changes.



a)



b)

Fig. 3. Application interface in different testing modes: LED testing (a), photodetector testing (b)

Of particular note is the DEBUG modal window, which can be opened via a dedicated button. This window is designed for low-level hardware control and allows manual adjustment of digital potentiometer values, control of the transistor and illuminating LED, as well as real-time monitoring of all measured voltages.

Overall, the developed graphical interface combines the functionality of professional measurement equipment with ease of use. The modular code structure and clear separation of logic and presentation make it easy to add new features and adapt the application for specific needs [7].

After the development and debugging of the basic version of the measurement system, a detailed analysis of its performance was conducted. For this purpose, clear evaluation criteria were defined, and appropriate test components were selected.

The main criteria for assessing the system's performance were: the total time required to complete a full cycle of *I-V* characteristic measurements, the number of measured points and their distribution across the voltage range, the resolution in critical regions of the characteristic, and the quality of the obtained data, including smoothness of the graph, absence of artifacts and jumps, and monotonic voltage increase.

To objectively evaluate the system, three different types of components with fundamentally different characteristics were used: a 1 kΩ resistor, a general-purpose white LED, and a BPW21R photodiode. All measurements were performed under identical parameters to ensure a valid comparison. For the resistor and the LED, the voltage range was −2 V to +5 V with a maximum allowable current of 15 mA. For the photodiode, a wider range (−5 V to +5 V) was used due to its operation under reverse bias, with a maximum current of 5 mA and illumination set to 100% (maximum brightness of the white LED illuminator). All measurements were carried out on the same hardware platform without changes in circuitry between tests.

Based on the test results, several shortcomings of the basic system version were identified. These can be grouped into three main categories: slow performance, inefficient distribution of measurement points, and insufficient data quality.

The most obvious problem was the low operating speed of the system. A detailed analysis of the timing characteristics showed that each measurement point takes approximately two seconds, including:

- 0.5 seconds – delay after sending the potentiometer position change command (DELAY_COMMAND);
- 0.5 seconds – delay before and after the measurement for system stabilization (DELAY_MEASURE);
- 1 second – time required for data transmission via the serial port and processing.

It is important to note that these delays are not excessive – they were determined experimentally as the minimum values at which the system operates stably without failures. Shorter delays led to unstable measurements due to insufficient time for voltage stabilization after changing the potentiometer positions or to missed commands due to the speed limitations of the serial interface. In contrast, the real source of slowness is the excessive number of measurements. The system performs a measurement at every potentiometer step, even when it is clearly unnecessary.

The second issue is related to the distribution of points across the voltage range. The basic version of the system uses a fixed step size for potentiometer changes, which results in an excessively dense placement of points in linear regions. This problem is especially pronounced during adjustments of pot2, which regulates resistance in the current circuit. This potentiometer has a range from 0 to 127, and the system sequentially passes through all these values with a step of 1. However, most of this range corresponds to the dead zone or the pre-bend region of the LED and the linear element, where detailed scanning is unnecessary.

The third group of problems directly concerns the quality of the obtained measurements. Analysis of the graphs revealed several systematic artifacts:

- *Points with reverse voltage movement*: This may be caused by ADC reference voltage instability, measurement noise due to coupling from potentiometers or other sources, or inaccuracies in setting the potentiometers due to their discrete nature. Such points create small "spikes" on the graph;

- *Jumps during range switching*: When the system switches the measurement range (turns on transistor VT1), a small current jump inevitably occurs. Although the algorithm attempts to select a post-switch voltage higher than the previous one, due to the discreteness of the potentiometers and ADC, it is impossible to perfectly smooth this transition. As a result, a small gap in the current is visible on the graph;
- *Insufficient noise filtering*: When working with low currents, the relative measurement noise can reach 5–10%. To filter such noise, the basic version averages 30 ADC measurements per point.

All of these identified shortcomings became the basis for defining the requirements for an optimized version of the system, which was expected to provide:

- a significantly higher measurement speed by reducing the number of redundant measurements;
- a rational distribution of points, with detailed scanning of critical regions and rapid traversal of linear sections;
- improved data quality through artifact filtering and ensuring the monotonic change of voltage.

Strategy of adaptive optimization of measurements

The analysis of the basic system's limitations clearly indicated that the primary issue is the excessive number of measurements in regions that provide little or no useful information. To address this, an adaptive algorithm was implemented, which automatically adjusts the measurement density according to the segment of the *I-V* characteristic under investigation.

The core idea of the adaptive approach is to perform a local analysis of the *I-V* characteristic in real time during the measurement process [16]. Rather than proceeding with a fixed step across the entire range, the system evaluates several of the most recently measured points and, based on this analysis, determines whether the next measurement should be taken with a small step or whether several potentiometer values can be skipped.

To implement this approach, the concept of the derivative – the rate of change of one quantity relative to another – was employed. For *I-V* characteristic, two options are possible: *dI/dV*, representing the differential conductance, and *dV/dI*, representing the differential resistance. The choice depends on the parameter being controlled at a given stage of the algorithm: when the voltage across the element is varied (*pot1*), *dI/dV* is analyzed; when the circuit resistance is varied (*pot2*), *dV/dI* is more informative.

The key concept is to classify the current region of the characteristic based on the local derivative, estimated by a first-order finite difference, i.e., the slope of the secant line between adjacent points. Small derivatives (slow changes in current or voltage) indicate "uninteresting" regions, where large steps can be taken, whereas large or rapidly changing derivatives mark critical regions requiring fine-grained measurements.

To calculate the derivative, the system stores a history of the last N measurements (typically 3–10). Local derivatives are computed for each pair of adjacent points, with mean and maximum values derived across the history. These metrics assess the rate of parameter change and identify sharp transitions indicative of key points.

Based on this analysis, the system determines the appropriate step size (number of potentiometer units to the next measurement) and whether measurements can be skipped. Importantly, the adaptive algorithm does not alter the core logic of the basic system; it optimizes the scanning trajectory while preserving all safety checks, current and voltage limits, and the overall measurement sequence.

Let us examine these two modes in more detail. The first mode of adaptive analysis (*dI/dV* mode) is applied at stages of the algorithm where the system varies the voltage across the element under study. This occurs when adjusting *pot1* (anode voltage) or *pot3*

(cathode voltage). In these cases, it is logical to analyze how the current responds to changes in voltage. The derivative of current with respect to voltage characterizes the differential conductance of the element, which, for a resistor, is inversely proportional to its resistance and remains constant across the entire characteristic. For semiconductor elements, this quantity depends strongly on the operating point and can vary by several orders of magnitude.

For LEDs, characteristic regions can be distinguished:

- *dead zone* (voltage below the threshold, approximately 0-2.3 V for a white LED): the current is nearly zero regardless of voltage, and *dI/dV* is close to zero. This region provides no useful information about the LED properties, so large steps can be taken here;
- *I-V characteristic bend* (threshold voltage region, approximately 2.4-2.8 V): the current begins to increase sharply and exponentially. *dI/dV* is very high and changes rapidly. This is the most critical region of the characteristic, requiring maximum measurement detail;
- *linear region* (after the bend, approximately 2.9-3.5 V): the current–voltage relationship approaches linearity, *dI/dV* is large but roughly constant. Moderate point density is sufficient in this region.

The derivative was calculated using the finite difference method. If two adjacent points in the history have coordinates (V_1, I_1) and (V_2, I_2), the local derivative is calculated as:

$$\frac{dI}{dV} = \frac{|I_2 - I_1|}{|V_2 - V_1|}, \tag{2}$$

where V_1 is the voltage of the previous measurement, I_1 is the current of the previous measurement, V_2 is the voltage of the current measurement, and I_2 is the current of the current measurement. Since the magnitude of the change, rather than its sign, is important, absolute differences are used.

This method was chosen for several reasons. First, it is extremely simple to implement. Since the calculations are performed in real time during measurements, more complex methods could slow down the system's operation.

Second, the finite difference method works well under our conditions. The distance between adjacent points is small enough for the local approximation to be valid. For example, when *pot1* is adjusted by one unit, the voltage changes by approximately 0.05-0.1 V, which is only 1-2 % of the full range. Within such a small range, most characteristics can be considered nearly linear.

Third, the method does not require complex assumptions about the nature of the function under study. Some alternative methods (e.g., polynomial or spline interpolation [17]) require knowledge of the function's smoothness or the availability of a certain number of derivatives. The algorithm, however, operates on real measurements, which may contain noise and discontinuities or non-smooth regions.

To improve the reliability of the estimation, the derivative is calculated for all pairs of adjacent points in the history. The mean derivative (*avg_derivative*) is then computed, characterizing the typical rate of current change in the current region, and the maximum derivative (*max_derivative*) is determined, which may indicate sharp transitions. Based on these computed derivatives, the system classifies the current region according to the logic presented in **Table 1**.

The second mode of adaptive analysis is used at stages where the system changes the resistance in the current path through the element. This occurs when adjusting *pot2*, which is connected in series with the element under study. Changing the position of the potentiometer causes a change in the current through the element, and consequently, a change in the voltage across it.

Table 1. **Description of the system's logic for classifying the currently studied regions**

Logic for zone determination	Explanation
Algorithm stage where the voltage across the element is changed (dI/dV mode)	
IF avg_derivative <0.001 A/V: Zone: "dead_zone" Step: 5 (large) Skipping allowed: yes	The threshold of 0.001 A/V was selected based on the consideration that, for typical voltage changes of 0.1 V in the LED's dead zone, the current changes by less than 0.1 mA, and *dI/dV* < 0.001. A large step allows the system to quickly traverse this "uninteresting" region.
IF max_derivative > avg_derivative × 2: Zone: "knee" Step: 1 (minimum) Skipping allowed: not	This condition detects sharp jumps in the derivative. If the maximum value is twice the average, the history contains points with very different rates of current change – a characteristic sign of a bend. The experimentally chosen factor of 2 represents a compromise between sensitivity and stability of the classification.
IF avg_derivative < 0.05 A/V: Zone: "linear" Step: 3 (middle) Skipping allowed: yes	The threshold of 0.05 A/V separates moderately increasing current from rapidly increasing current. In the linear region, the current rises predictably, so an average step can be used, and measurements can occasionally be skipped.
ELSE: Zone: "active" Step: 5 (large) Skipping allowed: not	If none of the previous conditions are met, the current increases rapidly (*dI/dV* > 0.05) but without sharp jumps. This is the operating region of the LED, which does not require very detailed scanning.
Algorithm stage where the resistance in the current circuit through the element is changed (dV/dI mode)	
IF avg_resistance > 100 Ω: Zone: "high_resistance" Step: 5 (large) Skipping allowed: yes	The threshold of 100 Ω corresponds to the LED's dead zone or the region of very low photodiode currents. At such a high resistance, changing the position of *pot2* has virtually no effect on the current through the element, so this region can be traversed quickly.
IF resistance_spread > avg_resistance × 3: Zone: "resistance_transition" Step: 1 (minimum) Skipping allowed: not	This condition detects resistance instability. If the range of values is three times greater than the average, it indicates a sharp change in the behavior of the dependence – a sign of a transition between the operating modes of the element.
IF avg_resistance > 10 Ω: Zone: "medium_resistance" Step: 3 (middle) Skipping allowed: yes	The range of 10-100 Ω corresponds to the region where the LED is partially turned on but not yet in full active mode. Moderate measurement detail is required in this region.
ELSE: Zone: "low_resistance" Step: 5 (large) Skipping allowed: not	For resistances below 10 Ω, the element operates in the active mode with high currents. This region does not require very detailed investigation.

The derivative of voltage with respect to current (dV/dI) represents the differential resistance of the element at a given point on the characteristic. For a resistor, this value is constant and equal to its nominal resistance. For semiconductor elements, it strongly depends on the operating point.

For LEDs, characteristic regions can be distinguished in terms of dynamic resistance:

- *high resistance* (thousands to tens of thousands of Ω – dead zone): current is nearly absent, and a large voltage change causes only a very small change in current;
- *transition region* (*I*–*V* curve bend): resistance changes sharply from high to low. This is the region of greatest instability in *dV/dI*;
- *moderate resistance* (tens to hundreds of Ω – linear region): resistance is relatively constant;
- *low resistance* (single Ω – saturation region): at high currents, the resistance decreases due to the conduction onset.

The calculation of dynamic resistance is performed similarly to the calculation of conductance, but with the order of quantities reversed:

$$\frac{dV}{dI} = \frac{|V_2 - V_1|}{|I_2 - I_1|}, \tag{2}$$

where V_1 is the voltage of the previous measurement, I_1 is the current of the previous measurement, V_2 is the voltage of the current measurement, and I_2 is the current of the current measurement.

An important feature is the need to check for division by zero or very small values. If the current changes by less than 1 μA (corresponding to the measurement accuracy), the derivative calculation is skipped for that pair of points.

As in the voltage mode, statistical characteristics are calculated: the average dynamic resistance (*avg_resistance*), the maximum and minimum resistance values in the history (*max_resistance* and *min_resistance*), and the spread of values (*resistance_spread* = *max_resistance* −*min_resistance*), which indicates stability. Classification in the current mode is based on the absolute resistance value and its variability (see **Table 1**).

The key issue in implementing the adaptive algorithm is determining the mode of analysis (voltage or current) to use at each measurement stage. This decision is based on the physical understanding of what is being controlled and what is being measured. If, at a certain stage of the algorithm, the position of *pot1* or *pot3* changes (voltage control) (at a certain stage of the algorithm), the *dI/dV* mode (voltage-based analysis) is used. In this case, changing *pot1* or *pot3* directly alters the potential at the anode or cathode of the studied element, meaning the voltage is controlled and the current response is measured.

If, however, the position of *pot2* changes (circuit resistance control), the *dV/dI* mode (current-based analysis) is used. Changing *pot2* alters the total circuit resistance, which causes a change in current, and the corresponding voltage change is measured.

It is important to understand that in both cases, the same *I*-*V* characteristic is being analyzed, just from different “perspectives.” The voltage mode corresponds to movement along the characteristic from left to right (*V* increases while observing *I*), while the current mode corresponds to movement from bottom to top (*I* increases due to a decrease in total resistance, while observing *V*).

Using different analysis modes for different stages provides several important advantages:

- *first*, each mode is optimally suited for its specific task. The voltage mode is ideal for detecting the bending of a LED’s *I*-*V* characteristic, where the current increases exponentially with increasing voltage, whereas the current mode is suitable for high-resistance regions, where small changes in current cause large changes in voltage;

- *second*, different stages of the algorithm have different “problematic zones”. For example, the *pot2* increase mode is particularly slow for photodiodes due to a long high-resistance region. The current mode with a 100 Ω threshold is perfectly suited for quickly passing through this region;
- *third*, a differentiated approach makes the system more versatile. Automatic mode selection allows effective operation with all optoelectronic components without changing settings.

However, the adaptive approach also has a number of limitations:

- *first*, a minimum measurement history is required for correct operation – at least 3 points. This means that the first few measurements at each stage are always performed with a small fixed step, regardless of the nature of the dependence;
- *second*, the classification of zones is based on thresholds (0.001; 0.01; 100 Ω, etc.). These values work well for typical LEDs and photodiodes but may be suboptimal for specific components. Therefore, it is necessary to conduct an investigation to determine the required step size and measurement accuracy of the system;
- *third*, the adaptive algorithm adds some complexity to the program logic and requires additional computations. Although these calculations are performed on the computer side and do not affect the Arduino’s operating speed, they still make the code more complicated to understand and configure.

The main method is skipping measurements in regions where the characteristic changes predictably. It is implemented through the *skip_allowed* parameter, which is returned by the analysis function together with *suggested_step*. When *skip_allowed* = *True*, the system can skip several consecutive measurements by increasing the potentiometer position without communicating with the Arduino. The number of skips is limited by the MAX_CONSECUTIVE_SKIPS parameter (for example, 5) to avoid missing important changes in the characteristic. Skipping is allowed only in the *dead_zone*, *linear*, *high_resistance*, and *medium_resistance* regions, where the characteristic changes smoothly. In the *knee*, *active*, *resistance_transition*, and *low_resistance* regions, skipping is prohibited because abrupt changes may occur. This strategy is especially effective for photodiodes, where the long horizontal section of the characteristic allows up to 5 consecutive measurements to be skipped, resulting in a time reduction of up to 50-60 % during reverse bias scanning.

In addition to speed optimization, several mechanisms for improving the quality of the acquired data were implemented. When adding a new point to the graph, the monotonic increase of voltage is verified. Statistics of the filtered points show that typically 2-5 % of points are discarded, which noticeably improves the smoothness of the graphs.

To objectively evaluate the effectiveness of the implemented optimizations, a series of comparative experiments was conducted in which the characteristics of the same test elements as in the baseline version of the system were measured. To ensure a fair comparison, all measurements were carried out on the same hardware without any changes to the circuitry or hardware configuration. The system was powered by a 9 V power supply, ensuring stable voltage conditions. The room temperature was maintained at 20-22 °C to minimize temperature drift in the characteristics of the semiconductor elements.

Each of the components was tested ten times in both the baseline and the optimized versions of the system. Such a sample size makes it possible to obtain statistically significant results and evaluate not only the average values but also the parameter variation. A pause of 1-2 minutes was maintained between tests to allow the system to stabilize.

For each test, the following parameters were recorded: the total test execution time from start to finish, measured with one-second precision; the number of points added to the *I–V* characteristic graph; the distribution of points across different regions of the characteristic (qualitative assessment); and the visual quality of the resulting graph (curve smoothness and presence of artifacts). From the ten measurements for each component, the mean value and standard deviation were calculated for both time and number of points. This made it possible to evaluate not only the typical performance but also the stability of the system operation.

As in the baseline version, the optimized version of the system used the same three components: a 1 kΩ resistor, a general-purpose white LED with a threshold voltage of approximately 2.5 V, and a BPW21R photodiode, as well as the same testing parameters described above.

Before testing began, optimal parameters for the *AdaptiveController* class were selected based on a preliminary analysis of the component characteristics. The tuning process involved several iterations of testing with different threshold values, followed by analysis of the point distribution and zone classification.

In particular, the final parameters for the voltage mode (*dI/dV* analysis) were selected as follows:

- the dead-zone threshold was set very low (0.001 A/V);
- the linear-zone threshold was set to 0.05 A/V. This value works well for 1 kΩ resistors, where $dI/dV \approx 1/R = 1$ A/V, which is higher than the defined threshold. Therefore, resistors are classified as being in the active region, resulting in a step size of 5 instead of a smaller step, which proved to be an optimal compromise;
- the bend detection multiplier was reduced to 2, making the system more sensitive to abrupt changes in the derivative. This is critically important for photodiodes, where the transition from reverse to forward bias may exhibit only a relatively moderate increase in the derivative;
- the step size for the dead zone was increased to 5, allowing the system to traverse “uninteresting” regions very quickly. For the linear and active regions, the step sizes were set to 3 and 5 respectively, representing a compromise between speed and detail.

In turn, the final parameters for the current mode (*dV/dI* analysis) were selected as follows:

- the high-resistance threshold was set to 100 Ω, which effectively separates the dead zone of photodiodes (where resistance may reach the kΩ range) from active regions;
- the transition detection multiplier was set to 3, which proved sufficient for detecting abrupt resistance changes without excessive sensitivity to noise;
- the step sizes for the different resistance regions were set to 5, 3, and 5 for high, moderate, and low resistance respectively, ensuring a balance between speed and safety.

The history size for the analysis was set to MIN_HISTORY = 3 and MAX_HISTORY = 5, meaning that the system analyzes the last 3-5 measurements before making a decision. A smaller value would make the system overly sensitive to noise, while a larger value would make it too inert to characteristic changes. The maximum number of consecutive skipped measurements (MAX_CONSECUTIVE_SKIPS) was set to 5. Thus, even in the most stable regions, the system is forced to perform a контрольне measurement after every five skipped points. The reverse-voltage movement filtering tolerance (VOLTAGE_TOLERANCE) was set to 5 mV, corresponding to approximately one ADC quantization step and effectively filtering noise without losing useful data.

These parameters demonstrated the best balance between speed, quality, and universality during testing across all three types of components.

A *1 kΩ resistor* is an ideal component for evaluating the efficiency of the system when working with linear elements. **Fig. 4** shows its I–V characteristic in the baseline (**Fig. 4a**) and optimized (**Fig. 4b**) versions of the system.

The analysis of the point distribution revealed non-uniformity: in the region from -5 V to +0.5 V, a clustering of points was observed, whereas in the range from +3 V to +5 V, the number of points was slightly lower (**Fig. 4a**). This non-uniformity is related to the operating characteristics of digital potentiometers and the nonlinear dependence of voltage on their position.

Despite the significant reduction in detail in the optimized version, the quality of the characteristic remained high – the straight-line behavior is reproduced with sufficient accuracy for practical applications (see **Fig. 4b**). The slight increase in time per point (1.53 sec in the baseline version versus 1.76 sec in the optimized version) is explained by the additional time required for derivative calculations and classification. However, this increase in algorithmic complexity is fully compensated by the reduction in the number of measurement points.

The analysis of the adaptive controller operation showed that the resistor is consistently classified as an “active region” with $dI/dV \approx 1.0$ A/V. When changing *pot2*, the system switches to current-analysis mode and classifies the characteristic as a “low-resistance” region.

The point distribution (**Fig. 4b**) did not become more uniform – the system displays approximately the same spread of points as in the baseline version, with clustering in the range from -0.5 V to +0.5 V. Visually, the graph of the optimized version does not differ from the baseline one – both versions produce nearly ideal straight lines.

The *white LED* represents the most interesting case for the adaptive algorithm due to the clearly distinguishable regions of its characteristic: the dead zone, the knee region, and the active region. **Fig. 5** presents its *I–V* characteristic in the baseline (**Fig. 5a**) and optimized (**Fig. 5b**) versions of the system.

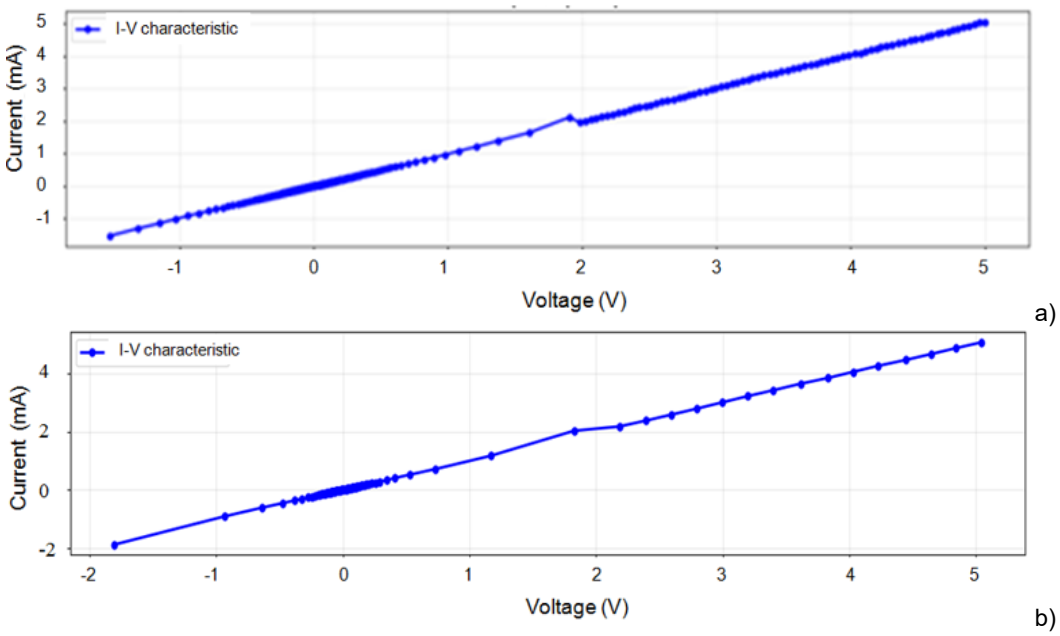


Fig. 4. Current–voltage characteristic of a 1 kΩ resistor in the baseline (a) and optimized (b) versions of the system

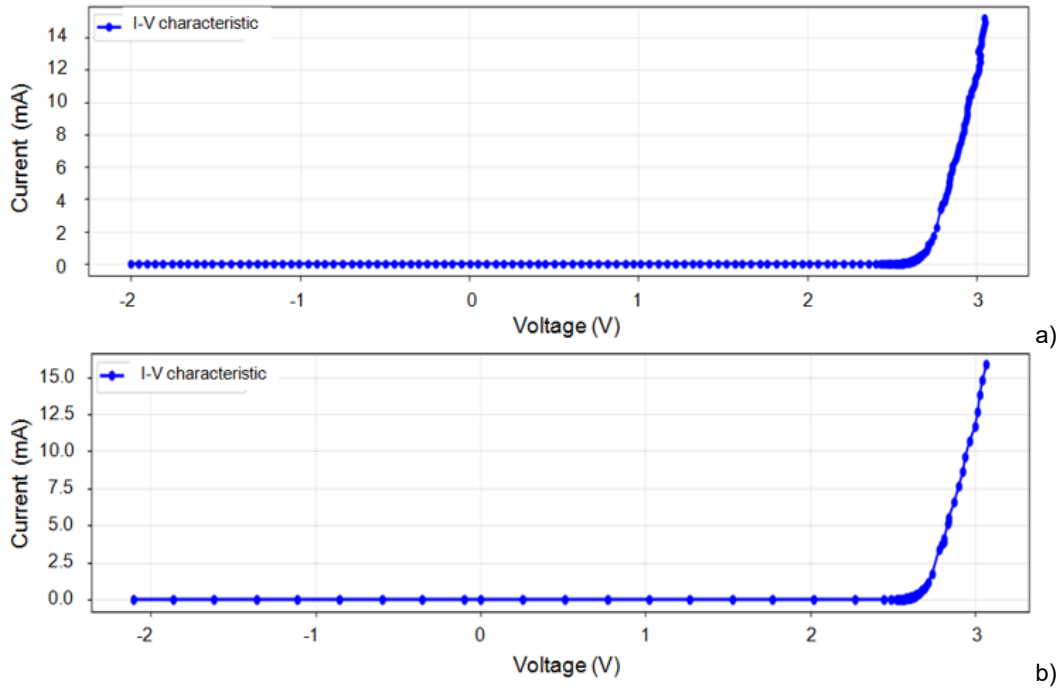


Fig. 5. Current–voltage characteristic of a white LED in the baseline (a) and optimized (b) versions of the system

The point distribution in the baseline version of the system was approximately as follows: 90 points in the dead zone (0-2.3 V), 200 points in the knee region of the *I–V* characteristic (2.3-2.7 V), and 65 points in the active region (2.7-3.1 V). The main issue was the excessive number of points in the dead zone, where the current is practically zero and the characteristic contains no useful information, as well as the excessive number of points in the knee region of the *I–V* curve (**Fig. 5a**).

The point distribution in the optimized version of the system, shown in **Fig. 5b**, was approximately 20 points in the dead zone, 70 points in the knee region, and 15 points in the active region. Critically, the number of points in the knee region remained sufficient for detailed reproduction of the exponential dependence of current on voltage.

Analysis of the obtained results revealed an almost threefold acceleration of the test execution while reducing the number of points to 29.5 % of the baseline version. The key achievement is the preservation of high characteristic quality in critical regions.

The analysis of region classification demonstrated the correct operation of the algorithm:

- in the dead zone (V < 2.3 V), the system stably detects dI/dV < 0.001 A/V and uses a step size of 5 with permission to skip points;
- in the knee region of the *I–V* characteristic (V ≈ 2.3-2.8 V), the system detects a sharp increase in the derivative (max_derivative > avg_derivative × 2) and switches to a no-skipping mode. This ensures detailed scanning of the most important region of the characteristic;
- in the active region (V > 2.9 V), the current increases rapidly but more or less linearly, and the system switches to the high-current measurement mode.

Another positive effect is the reduction in the number of filtered points associated with reverse voltage movement. In the baseline version, an average of 12-15 points (3-5 % of the total number) were filtered out, whereas in the optimized version only 1-3

points (1-3 %) were removed. This indicates that the adaptive algorithm is less prone to creating situations involving accidental voltage decreases.

The *BPW21R photodiode* is the most problematic component for measurement due to the very long horizontal region of its characteristic under reverse bias. **Fig. 6** presents its *I–V* characteristic in the baseline (**Fig. 6a**) and optimized (**Fig. 6b**) versions of the system.

The main problem of the baseline version is the enormous number of redundant points on the horizontal region, where the current is practically constant at approximately 0.9 mA (see **Fig. 6a**). These points provide no additional information.

The observed reduction in both absolute measurement time and point dispersion indicates more predictable operation of the optimized system. Analysis of the results revealed more than a twofold acceleration of the test execution while reducing the number of points to 42.0 % of the baseline version (**Fig. 6b**).

The analysis of the algorithm operation demonstrated effective region classification:

- on the horizontal region under reverse bias, the system detects a very small derivative *dI/dV* (practically zero) and a high dynamic resistance (> 100 Ω) in current mode, and therefore enables point skipping;
- in the transition region (+0.1...+0.5 V), the system correctly detects an increase in the derivative and reduces the step size to 1-3, ensuring detailed scanning.

An important result is that even with the number of points reduced by more than half, the quality of the characteristic did not deteriorate (see **Fig. 6b**). Visual comparison of the graphs shows practically identical curves with the same level of detail in the critical regions.

A comparison of the testing results for the electronic components investigated is summarized in **Table 2**.

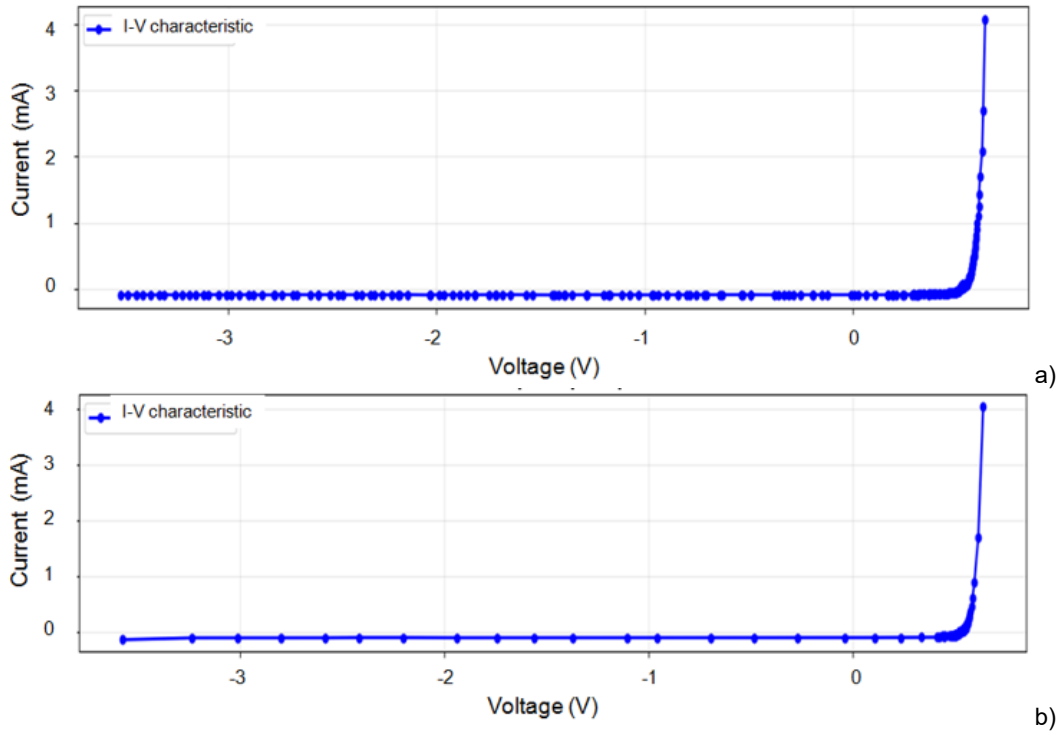


Fig. 6. Current–voltage characteristic of the BPW21R photodiode in the baseline (a) and optimized (b) versions of the system

Table 2. **Results of testing electronic components in the baseline and optimized versions of the system**

Component	The baseline version				The optimized version				Speedup, times	Point savings, %
	Test duration, sec	Point number	Duration standard deviation	Number point standard deviation	Test duration, sec	Point number	Duration standard deviation	Number point standard deviation		
1 kΩ resistor	735	480	4.3	1.8	172	98	3.1	2.1	4.27	79.6
White LED	458	356	4.8	1.9	155	105	3.2	2.3	2.95	70.5
BPW21R photodiode	675	355	4.9	2	285	149	3.3	2.8	2.37	58

The developed adaptive optimization system demonstrates a stable speedup of 2.4-4.3 times depending on the type of component, while maintaining high-quality characteristics. The greatest performance gain is observed for components with predictable behavior.

Importantly, the optimization does not degrade data quality – critical regions are scanned with high resolution, while redundant measurements in predictable regions are eliminated. Visual comparison of the graphs shows that the optimized version does not worsen the *I–V* characteristics compared to the baseline version, which is explained by better filtering and the absence of certain artifacts present in the baseline version. The reverse-voltage movement filter effectively removes “jagged edges” from the graph.

For the LED, the *I–V* curve knee in the optimized version is reproduced somewhat better than in the baseline due to the concentration of points in this region. The exponential increase in current from 0.1 mA to 15 mA with a voltage change from 2.5 V to 3.1 V is clearly visible in both graphs, but the optimized version has a higher density of points specifically in the initial part of the knee (2.3-2.7 V), where the rate of change is maximal. For the photodiode, the transition region from reverse to forward bias (+0.1 V to +0.5 V) in the optimized version has sufficient detail for correctly reproducing the nature of the dependence.

Overall, the qualitative assessment shows that the optimized version is not only faster but also provides even better measurement quality compared to the baseline version. This confirms the success of the chosen optimization approach.

CONCLUSION

A fully functional system for automated measurement of current-voltage and optical characteristics of optoelectronic devices has been developed and implemented based on the Arduino UNO microcontroller with adaptive optimization of the scanning process. The system combines hardware built from readily available electronic components with software for both the microcontroller and a personal computer.

The baseline version of the system with a fixed scanning step demonstrated reliable operation for all types of investigated components; however, performance analysis revealed significant drawbacks that led to excessively long test durations. To address the issue of inefficient time usage, an adaptive optimization system based on numerical analysis methods was developed. The system was integrated into the existing code with minimal

changes to the core logic, preserving all safety and calibration functions. Additionally, a reverse-voltage movement filter was implemented, which removes artifact points caused by measurement noise, improving the visual quality of the obtained characteristics.

An experimental study of the optimized system's efficiency was carried out on three types of electronic components, with ten independent measurements performed for each type. The developed adaptive optimization system demonstrates a stable speedup of 2.4-4.3 times depending on the component type while maintaining high-quality characteristics. Importantly, during all thirty experiments, no cases were recorded in which critical regions were missed or the characteristic shape was significantly distorted. This confirms the reliability of the developed region classification system and the correctness of the chosen adaptive algorithm parameters. It should also be noted that the measurement time per point increases by 10-15 % due to additional derivative and classification computations; however, this is fully compensated by the reduction in the total number of points. The region classification parameters were selected empirically, so specific components may require adjustment of threshold values.

Overall, the presented work demonstrates the possibility of creating an efficient and accessible system for studying the characteristics of optoelectronic devices based on modern microcontroller platforms. The developed adaptive optimization system has proven effective under real operating conditions, providing a significant increase in measurement speed while maintaining high result quality.

Comparison with simpler adaptive scanning strategies was not conducted, and this remains a subject for future research.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [M.R., Yu.K]; methodology, [M.R., Yu.K., B.K.]; investigation, [M.R., B.K.]; writing – original draft preparation, [M.R., B.K.]; writing – review and editing, [M.R., Yu.K., B.K.]; visualization, [M.R., Yu.K., B.K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Piprek J. (2018). Handbook of Optoelectronic Device Modeling and Simulation. Vol. II. Lasers, Modulators, Photodetectors, Solar Cells, and Numerical Methods. *CRC Press.* 233 p. http://lab.semi.ac.cn/library/upload/files/2020/8/59160607.pdf

[2] Barua P. (2023). Optoelectronics: Basics and Applications. *Kindle Edition.*104 p. https://www.amazon.com/Optoelectronics-Basics-Applications-Prasun-Barua/dp/B0CD15SBN7

[3] Monk S. (2017). Hacking Electronics: Learning Electronics with Arduino and Raspberry Pi. *McGraw Hill TAB*. 276 p. https://rpitst.com/img/ebook/1710996146_7958c4023e1d82a99bd4.pdf

[4] Arsath Natheem S. (2021). Handbook of Arduino: 100+ Arduino Projects learn by doing practical guides for beginners and inventors. *Kindle Edition*. https://www.scribd.com/document/721172292/Handbook-of-Arduino-100-Arduino-Projects

[5] Burden R. L., Faires J. D. (2010). Numerical Analysis (9th ed.) *Brooks/Cole*, https://faculty.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf
[6] Cejer M. A., Tucker J. L., Weng L. (2013). Choosing the Optimal Source Measurement Unit Instrument for Your Test and Measurement Application. *Keithley*. p. 1-18. https://download.tek.com/document/HowToCHooseSMU.pdf
[7] Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. (2014) Best Practices for Scientific Computing. *PLoS Biol* 12(1): e1001745. https://doi.org/10.1371/journal.pbio.1001745
[8] Python Software Foundation. (2026). *tkinter — Python interface to Tcl/Tk*. Python 3 documentation. https://docs.python.org/3/library/tkinter.html
[9] PySerial Documentation. (2020). *PySerial*. https://www.pyserial.com/docs
[10] Hunter J., Dale D., Firing E., Droettboom M., the *Matplotlib* Development Team (2017) Matplotlib. https://matplotlib.org/2.1.0/Matplotlib.pdf
[11] Matplotlib 3.10.8 documentation. (2025). *Mtplotlib*. https://matplotlib.org/stable/
[12] Tosi S. (2009). Matplotlib for Python Developers. *Packt Publishing*, https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/Python/Matplotlib%20for%20Python%20Developers%20%282009%29.pdf
[13] Introduction to Threading in Python. (2025). *ManageEngine Site24x7*. https://www.site24x7.com/learn/threading-in-python.html
[14] EasyEDA Tutorial. V.6.4.32. (2022). https://image.easyeda.com/files/EasyEDA-Tutorial_v6.4.32.pdf
[15] MCP413X/415X/423X/425X: Single/Dual Digital Potentiometer with SPI Interface. Datasheet. (2008). *Microchip Technology Inc.* https://www.lcsc.com/datasheet/C640354.pdf
[16] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (2007). Numerical Recipes: The Art of Scientific Computing (3rd ed). *Cambridge University Press.* 1235 p. https://assets.cambridge.org/97805218/80688/frontmatter/9780521880688_frontmatter.pdf
[17] Karpfinger, C. (2022). Polynomial and Spline Interpolation. In: Calculus and Linear Algebra in Recipes. *Springer, Berlin, Heidelberg*. p. 311–320. https://doi.org/10.1007/978-3-662-65458-3_29

ОПТИМІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДОСЛІДЖЕННЯ ДИСКРЕТНИХ ОПТОЕЛЕКТРОННИХ ПРИСТРОЇВ З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРА ARDUINO

Михайло Русиняк, Юрій Корчак*, Богдан Кужель
Кафедра оптоелектроніки та інформаційних технологій
Львівський національний університет імені Івана Франка
вул. Ген. Тарнавського 107, м. Львів, 79017, УКРАЇНА

АНОТАЦІЯ

Вступ. Якісне дослідження електричних і світлових характеристик оптоелектронних компонентів є необхідною складовою під час проєктування,

розробки та тестування оптоелектронних систем. З цією метою розроблено систему автоматизованого вимірювання характеристик оптоелектронних пристроїв на базі мікроконтролера Arduino UNO з програмованим керуванням напругою через цифрові потенціометри та передачею даних на комп'ютер для візуалізації, а також проведено її оптимізацію.

Матеріали та методи. Повнофункціональна система автоматизованого вимірювання вольт-амперних та світлових характеристик оптоелектронних пристроїв розроблена та реалізована на базі мікроконтролера Arduino UNO. Апаратна частина системи побудована на базі цифрових потенціометрів MCP4131 з використанням інтегрованого люксметра GY-302, настільний додаток створено на мові програмування Python з використанням бібліотек Tkinter, pySerial, Matplotlib, Threading. Адаптивна оптимізація системи проведена на основі методу скінченних границь для обчислення відповідних локальних похідних.

Результати. Експериментальне дослідження ефективності оптимізованої системи показало суттєве підвищення продуктивності за збереження високої якості характеристик. Розроблена система адаптивної оптимізації демонструє стабільне пришвидшення в 2,4-4,3 рази залежно від типу елемента. Жодного разу під час проведених експериментів не було зафіксовано випадків пропускання критичних зон або суттєвого спотворення форми характеристики. Це підтверджує надійність розробленої системи класифікації зон та коректність вибору параметрів адаптивного алгоритму.

Висновки. Результати роботи демонструють можливість створення ефективної та доступної системи для дослідження характеристик оптоелектронних пристроїв на базі сучасних мікроконтролерних платформ. Розроблена система адаптивної оптимізації показала свою ефективність у реальних умовах експлуатації.. Це робить систему цінним інструментом як для навчальних, так і для практичних застосувань у галузі дослідження та розробки оптоелектронних пристроїв.

Ключові слова: вольт-амперна характеристика, метод скінченних різниць, мікроконтролер Arduino, світлодіод, фотодіод, цифровий потенціометр.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
30 June, 2026	15 September, 2026	21 September, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 205–218
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 681.518.3

RASPBERRY PI 5-BASED MULTICHANNEL MEASUREMENT SYSTEM FOR GAS-FLOW RESISTANCE AND TEMPERATURE MONITORING IN A NATURAL ZEOLITE PACKED BED

***Grzegorz Walowski*[1], *Dorota Anders*[1], *Barbara Dybek*[1], *Vadym Ptashnyk*[2], *Serhiy Syrotyuk*[2]*, *Patryk Radek*[3]**
[1] *Institute of Technology and Life Sciences – National Research Institute, 3 Hrabska Avenue, Falenty, 05-090 Raszyn, POLAND*
[2]*Faculty of Mechanics, Energy and Information Technologies, Stepan Gzhytskyi National University of Veterinary Medicine and Biotechnologies of Lviv, 1 Volodymyra Velykoho Street, Dubliany, 79083, UKRAINE*
[3]*Department of Optoelectronics, Faculty of Electrical Engineering, Silesian University of Technology, 2 Krzywoustego Str., 44-100 Gliwice, POLAND*

Walowski, G., Anders, D., Dybek, B., Ptashnyk, V., Syrotyuk, S., Radek, P. (2026). Raspberry PI 5-based Multichannel Measurement System for Gas-Flow Resistance and Temperature Monitoring in a Natural Zeolite Packed Bed. *Electronics and Information Technologies*, 35, 205–218. https://doi.org/10.30970/eli.35.15

ABSTRACT

Background. Experimental characterization of granular beds for sorption thermal energy storage requires simultaneous measurement of gas-flow resistance and temperature using flexible multichannel data acquisition. This study developed a Raspberry Pi 5-supervised measurement system for a natural clinoptilolite packed bed.

Materials and Methods. Experiments were performed in a thermally insulated 0.5 m³ reactor containing a 0.4 m clinoptilolite bed with a particle-size fraction of 0.5–1.0 mm. Oxygen was used as the working gas. Pressure signals were processed by a miniSTM32-based subsystem, while ten Pt1000 sensors, ADG732 multiplexers, and an AD7124-4 converter were used for temperature monitoring. Three repeated series were performed at eight gas-flow settings. Pressure-flow data were analyzed using mean values, sample standard deviation, and linear regression; temperature-pressure relationships were evaluated by linear regression and Pearson correlation. A stationary two-dimensional MATLAB model was used to assess the thermal response.

Results. Functional operation of the platform was verified, including a 41-point comparison of three pressure channels showing close inter-channel agreement. Pressure drop ranged from 7.93 to 99.79 kPa, while the mean value increased from 25.49 to 93.41 kPa. Linear regression showed a significant positive pressure-flow relationship (R^2 = 0.840, p = 0.0014). Temperature remained within 16.83–16.92 °C, with no significant dependence on pressure drop. MATLAB-calculated mean temperatures differed from experimental means by no more than 0.08%, and the calculated temperature increase due to viscous dissipation was below 0.005 °C.

Conclusions. The developed system enables integrated measurement of gas-flow resistance and temperature in a natural clinoptilolite packed bed. Experimental and numerical results indicate that substantial changes in flow resistance were not accompanied by a detectable systematic temperature response under the investigated conditions. The architecture provides a basis for further studies of zeolite beds under controlled sorption conditions relevant to thermal energy storage.

© 2026 Grzegorz Walowski et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Keywords: embedded measurement system, Raspberry Pi 5, data acquisition, gas-flow resistance, natural zeolite, packed bed, thermal energy storage.

INTRODUCTION

The increasing integration of intermittent renewable energy sources and the need for more efficient use of industrial waste heat have intensified the demand for thermal energy storage technologies capable of decoupling heat generation from consumption. Among sensible, latent, and thermochemical storage concepts, sorption-based thermal energy storage is particularly attractive for long-term applications because energy can be stored through reversible adsorption–desorption processes with relatively high storage density and limited heat loss during the storage period. Zeolites are among the most extensively investigated sorption materials owing to their microporous structure, high affinity for polar adsorbates such as water vapor, thermal stability, and cyclic adsorption capability [1–5].

Both synthetic and naturally occurring zeolites have been investigated for thermochemical heat storage. Synthetic zeolite 13X generally exhibits high water uptake and favorable sorption performance, whereas natural zeolites are of interest because of their availability and potentially lower material cost. Recent studies have shown that the practical performance of zeolite-based storage systems depends not only on intrinsic sorption properties but also on heat and mass transfer conditions within the granular bed [1–3]. Consequently, the characterization of natural zeolite beds under controlled gas-flow conditions remains relevant when such materials are considered for thermal energy storage reactors.

In packed-bed reactors, gas-flow conditions directly influence pressure loss, mass transfer, and the spatial and temporal evolution of the temperature field. Experimental and numerical studies of zeolite reactors have demonstrated the importance of airflow rate, humidity, reactor geometry, and axial position for the resulting thermal response [6, 7]. Pressure drop is also an essential reactor-design parameter because it determines flow resistance and affects the energy required to transport the working fluid through the bed. Its magnitude depends on particle size and shape, bed porosity and dimensions, fluid properties, and superficial velocity; therefore, reliable simultaneous measurements of gas flow and pressure at representative locations are necessary for adequate packed-bed characterization [8].

Temperature measurements provide complementary information on heat-transfer processes occurring inside a packed bed. Modern experimental studies increasingly employ spatially distributed temperature sensing rather than relying exclusively on inlet and outlet temperatures. Such measurements allow thermal fronts, temperature gradients, and reaction zones to be identified and related to operating conditions [9,10]. In zeolite sorption reactors in particular, high-resolution intra-reactor temperature profiling has recently been used to characterize reaction-zone propagation and its relationship with thermal output [10]. These results indicate that a multichannel measurement system capable of acquiring temperature data at several positions within the bed can provide substantially more information than a single-point temperature measurement.

The implementation of such experiments requires a data acquisition system capable of integrating heterogeneous measurement channels, including resistive temperature sensors and pressure transducers, with signal conditioning, analog-to-digital conversion, data logging, and subsequent processing. Single-board computers such as the Raspberry Pi can serve as a supervisory acquisition and processing platform, while dedicated analog front-end circuits and microcontroller-based modules perform sensor-specific signal conditioning and low-level measurement tasks [11–15]. Such a distributed architecture provides flexibility for multichannel experiments and facilitates the integration of different sensor interfaces within a common measurement environment.

Despite considerable progress in both zeolite-based thermal storage characterization and embedded data acquisition technologies, these two research directions are commonly addressed separately. Experimental studies of granular sorption beds require simultaneous information on gas-flow resistance and temperature behavior,

whereas the acquisition of spatially distributed data demands a flexible multichannel measurement architecture. Integrating these functions within a dedicated embedded platform can facilitate experimental characterization of zeolite beds and provide a basis for further monitoring and control of thermal storage systems.

Therefore, the aim of this study was to develop and experimentally evaluate a multichannel embedded measurement system, supervised by a Raspberry Pi 5, for the characterization of a natural zeolite packed bed intended for thermal energy storage applications. The system integrates pressure and temperature measurement channels, dedicated signal-processing electronics, and embedded data acquisition. The experimental study focused on gas-flow resistance and temperature behavior in a 0.5 m³ reactor filled with a 0.5–1.0 mm zeolite fraction. A MATLAB-based numerical model was additionally used for preliminary simulation and visualization of the temperature distribution.

MATERIALS AND METHODS

Experimental setup and test material

The experimental investigations were carried out using a thermally insulated cylindrical reactor with a nominal internal volume of 0.5 m³ (**Fig. 1**). The reactor was manufactured from AISI 304 stainless steel and had an internal diameter of 0.71 m, a maximum internal height of 1.25 m, and a wall thickness of 0.8 mm. The total mass of the reactor was approximately 20 kg. The reactor was installed on a mobile platform and thermally insulated to reduce heat exchange with the surroundings during the experiments.

The gas supply was introduced through a single 8 mm diameter inlet located at the bottom of the reactor, whereas the outlet of the same diameter was located in the upper cover. The pressure was monitored at the reactor inlet and outlet and at five levels along the bed height. At each internal measurement level, pressure readings were obtained from four circumferentially distributed pressure ports and averaged to characterize the local pressure conditions within the bed. The gas flow rate was measured using an Aalborg SMR1 glass-tube rotameter installed at the reactor outlet. For the 0–150 mm scale configuration, the manufacturer specifies an accuracy of ±2% of full scale and a repeatability of ±0.25%. Oxygen was supplied from a compressed-gas cylinder through a pressure regulator and used as the working gas during the permeability experiments.



Fig. 1. Experimental test stand with the thermally insulated 0.5 m³ cylindrical reactor used for gas-flow and temperature measurements in the zeolite packed bed

Natural clinoptilolite of volcanic origin from a Slovak deposit, with a particle-size fraction of 0.5–1.0 mm, was used as the granular bed material. During the reported experiments, the zeolite bed height was 0.4 m. The loose bulk density of the zeolite was 0.82 g cm^{-3}, while the geometrically determined interparticle bed porosity was $\varepsilon = 0.41$. Measurements of volumetric gas flow rate V', pressure drop ΔP, and temperature T were performed under single-phase gas-flow conditions. The data obtained were subsequently used to characterize the relationship between gas-flow rate and flow resistance and to analyze the thermal response of the zeolite bed.

Measurement system and data acquisition

A multichannel measurement and data acquisition system was developed with the Raspberry Pi 5 acting as the supervisory acquisition and processing platform. The system was organized into two main measurement branches: a pressure-measurement subsystem and a multichannel temperature-measurement subsystem. Sensor-specific signal processing was performed at the lower hardware level, whereas the Raspberry Pi 5 provided integration of the measurement channels, communication with peripheral modules, data acquisition, and subsequent data handling. A dedicated measurement board was designed, assembled, and commissioned for this purpose. The functional architecture of the developed measurement and data acquisition system is shown in **Fig. 2**. During functional testing, the power supply, communication interfaces, and operational stability of the hardware were verified. The measurement application provided the communication interfaces required by the implemented subsystems, including SPI for communication with the AD7124-4 converter and an asynchronous serial interface using the Modbus RTU protocol for communication with the miniSTM32-based pressure

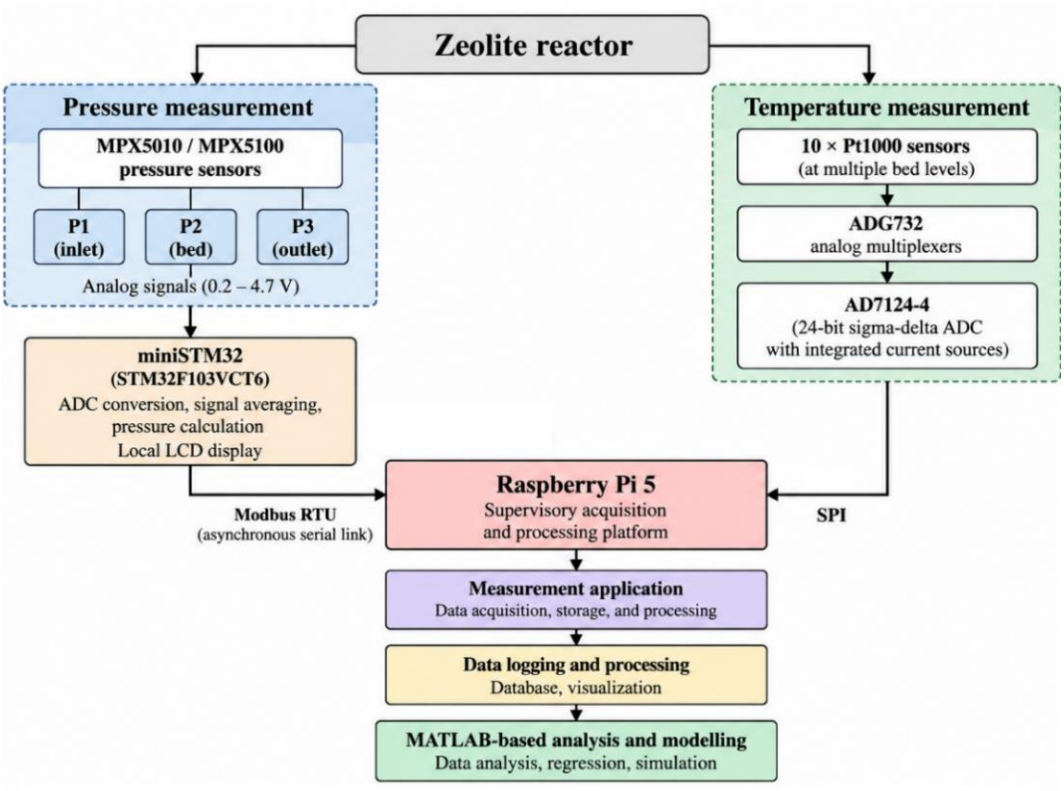


Fig. 2. Functional architecture of the embedded measurement and data acquisition system for pressure and multichannel temperature monitoring of the zeolite packed bed

subsystem. The three pressure-measurement channels were designated P1, P2, and P3, corresponding to the reactor inlet, the zeolite bed, and the reactor outlet, respectively.

The main components, their functions, and their integration within the measurement system are summarized in **Table 1**.

Semiconductor differential pressure sensors of the MPX5010 or MPX5100 series were used depending on the required measurement range (10 or 100 kPa). The sensors incorporate temperature-compensated piezoresistive sensing elements and integrated signal amplification. According to the manufacturer's specifications, the nominal transfer characteristics of the sensors can be expressed as:

$$V_{OUT} = V_S(kP + 0.004), \tag{1}$$

where V_{OUT} is the sensor output voltage, V_S is the supply voltage, P is the differential pressure, and k is the pressure-sensitivity coefficient equal to 0.09 kPa^{-1} for the MPX5010 and 0.009 kPa^{-1} for the MPX5100. The output-voltage tolerance specified for the sensor characteristic is ± 5% of the full-scale output span V_{FSS}, which is typically 4.50 V at V_S = 5.00 V.

Table 1. **Main components and functional organization of the measurement system**

Component / subsystem	Function	Configuration / interface	System integration
Pt1000 sensors	Temperature measurement	10 sensors: 1 at the inlet, 8 within the zeolite bed, and 1 at the outlet; three-wire connection	Connected to the temperature measurement chain through the ADG732 multiplexers and AD7124-4 converter
ADG732	Temperature-channel selection	Multichannel analog multiplexer	Provides sequential selection of Pt1000 channels for subsequent analog-to-digital conversion
AD7124-4	Pt1000 signal conditioning and analog-to-digital conversion	24-bit sigma-delta ADC; communication with Raspberry Pi 5 via SPI	Provides digital temperature data to the central acquisition platform
MPX5010 / MPX5100 pressure sensors	Differential pressure measurement	Measurement ranges of 10 and 100 kPa, respectively; three acquisition channels: P1, P2, and P3, analog output	Sensor signals supplied to the miniSTM32-based pressure-processing subsystem
miniSTM32 module	Processing of pressure-sensor signals	STM32F103VCT6-based subsystem; ADC conversion and signal averaging, Modbus RTU serial communication	Digitizes and processes pressure-sensor signals and transmits calculated pressure values to the supervisory acquisition system
Raspberry Pi 5	Centralized data acquisition and processing	Central computing platform; SPI and Modbus RTU serial communication	Integrates the temperature and pressure measurement subsystems and provides data logging and subsequent processing

For a supply voltage of 5.00 V, the nominal zero-pressure output voltage is 0.200 V. However, the experimentally determined zero-offset voltages of the sensors used in the measurement system were 0.182 V for the inlet channel, 0.176 V for the bed channel, and 0.182 V for the outlet channel. In addition, the sensor output signals were adjusted to the input-voltage range of the microcontroller ADC by a signal-level matching circuit. To account for the individual zero-offset deviations and the signal-conditioning stage, a corrected linear characteristic was determined separately for each pressure-measurement channel:

$$P = A \cdot U_{ADCIN} + B, \tag{2}$$

where P is the calculated pressure, U_{ADCIN} is the conditioned voltage applied to the ADC input, and A and B are channel-specific correction coefficients. The experimentally determined parameters of the three pressure-measurement channels are summarized in **Table 2**.

Pressure-sensor signals were processed using a miniSTM32 module based on the STM32F103VCT6 microcontroller. The three analog sensor signals were connected to the microcontroller ADC inputs, converted into digital form, averaged, and and subsequently converted to pressure values using the corrected channel characteristics given by Eq. (2). The calculated pressure values were displayed locally on the integrated LCD and transmitted to the supervisory data-acquisition system through an asynchronous serial link using the Modbus RTU protocol.

The temperature-measurement branch was based on ten Pt1000 platinum resistance temperature sensors. One sensor was installed at the reactor inlet and one at the outlet, while eight sensors were distributed vertically within the zeolite bed to obtain spatially resolved temperature information. The sensors operated in a three-wire configuration to reduce the influence of lead resistance. The Pt1000 measurement circuit used precision excitation current sources integrated into the AD7124-4 to convert sensor-resistance variations into voltage signals suitable for differential measurement and subsequent analog-to-digital conversion. Individual measurement channels were selected through ADG732 analog multiplexers, while signal conditioning and analog-to-digital conversion were performed using a 24-bit sigma-delta AD7124-4 converter interfaced with the Raspberry Pi 5 via SPI. The temperature acquisition circuitry was implemented as a dedicated shield for the Raspberry Pi platform, thereby providing a compact multichannel front end for monitoring the temperature distribution within the reactor.

At the system level, the Raspberry Pi 5 therefore acted as the common integration point for the pressure and temperature measurement branches. The pressure subsystem communicated with the supervisory platform through Modbus RTU, whereas the AD7124-4 temperature-conversion subsystem was interfaced through SPI. This separation between sensor-specific front-end electronics and the supervisory acquisition platform allows heterogeneous measurement channels to be incorporated into a single embedded system while retaining dedicated signal-processing paths for different sensor types.

Table 2. **Parameters of the corrected pressure-measurement channel characteristics**

Pressure channel	UOUT, V	UADCIN, V	A, Pa/V	B, Pa
P1 (inlet)	0.182	0.126	3209.877	−404.444
P2 (bed)	0.176	0.122	3205.829	−391.111
P3 (outlet)	0.182	0.127	3184.602	−404.444

Generalized cyber-resilience model synthesized from the literature

Gas-flow experiments were performed with the reactor filled with a 0.4 m layer of natural zeolite with a particle-size fraction of 0.5–1.0 mm. Oxygen was supplied to the bottom of the reactor through the pressure regulator, passed upward through the granular bed, and discharged through the outlet in the reactor cover. Three repeated measurement series were performed. Each series comprised eight gas-flow settings, for which the volumetric flow rate V', pressure drop ΔP, and temperature T were recorded. The same set of volumetric flow-rate values was used in all three series, enabling comparison of the repeatability of the measured pressure and temperature responses.

Prior to the packed-bed experiments, the agreement between the three pressure-measurement channels was evaluated by applying identical pressure conditions to the sensor inputs and comparing the calculated pressure values for channels P1, P2, and P3. The differences P1 – P2, P1 – P3, and P2 – P3 were determined for each test point. This procedure was used to assess inter-channel consistency after application of the individual correction characteristics.

The experimental data were analyzed in terms of two principal relationships: the dependence of pressure drop on volumetric gas-flow rate, $\Delta P = f(V')$, and the dependence of measured temperature on pressure drop, $T = f(\Delta P)$. For each of the eight gas-flow settings, the pressure-drop measurements from the three repeated measurement series were summarized using the arithmetic mean and sample standard deviation (SD, $n = 3$). The pressure-flow relationship was evaluated using linear regression applied to the mean pressure-drop values obtained at the eight gas-flow settings. The goodness of fit was assessed using the coefficient of determination R^2, and the statistical significance of the regression slope was evaluated using a two-sided test with a significance level of $p < 0.05$.

Because the measured temperature variations were small compared with the range of pressure changes, the $T = f(\Delta P)$ data were analyzed separately to determine whether a systematic relationship between the two variables could be identified. Linear regression and the Pearson correlation coefficient *r* were used to evaluate the direction and strength of the relationship. Statistical significance was assessed using a two-sided test with a significance level of $p < 0.05$. Statistical processing of the tabulated experimental data was performed using Python 3.13 with the NumPy and SciPy libraries.

In addition to the experimental data analysis, a stationary two-dimensional numerical model of energy and momentum balance in the porous zeolite bed was implemented in MATLAB. The model was used to evaluate whether the experimentally observed weak relationship between pressure drop and bed temperature was physically consistent with the expected thermal response of the system. The energy balance included heat generation associated with viscous dissipation of the gas flow and convective heat transfer through the reactor walls. The calculated mean bed temperatures were subsequently compared with the corresponding experimental mean temperatures obtained for the three measurement series.

RESULTS AND DISCUSSION

Measurement system implementation and functional evaluation

The proposed measurement architecture was implemented as a dedicated hardware and software platform for the experimental reactor. The custom measurement board was designed, assembled, and commissioned for operation with the Raspberry Pi 5. Initial verification included inspection of the assembled circuitry, replacement of faulty components where required, and functional testing of the power supply and communication interfaces. Particular attention was given to the SPI interface used by the AD7124-4 temperature-conversion subsystem, whose operation was successfully verified during commissioning. The miniSTM32-based pressure subsystem was also integrated

with the supervisory acquisition platform, including analog-to-digital conversion of the pressure-sensor signals, signal averaging, calculation of pressure values using the corrected channel characteristics, local visualization on the LCD, and data transmission through the Modbus RTU serial link.

The functional consistency of the three pressure-measurement channels was evaluated under identical pressure conditions. A total of 41 comparison points were recorded for channels P1, P2, and P3. As shown in **Fig. 3a**, the three channels followed the same imposed pressure profile over the complete set of 41 comparison points. Because the pressure values obtained from the individual channels were very close, the

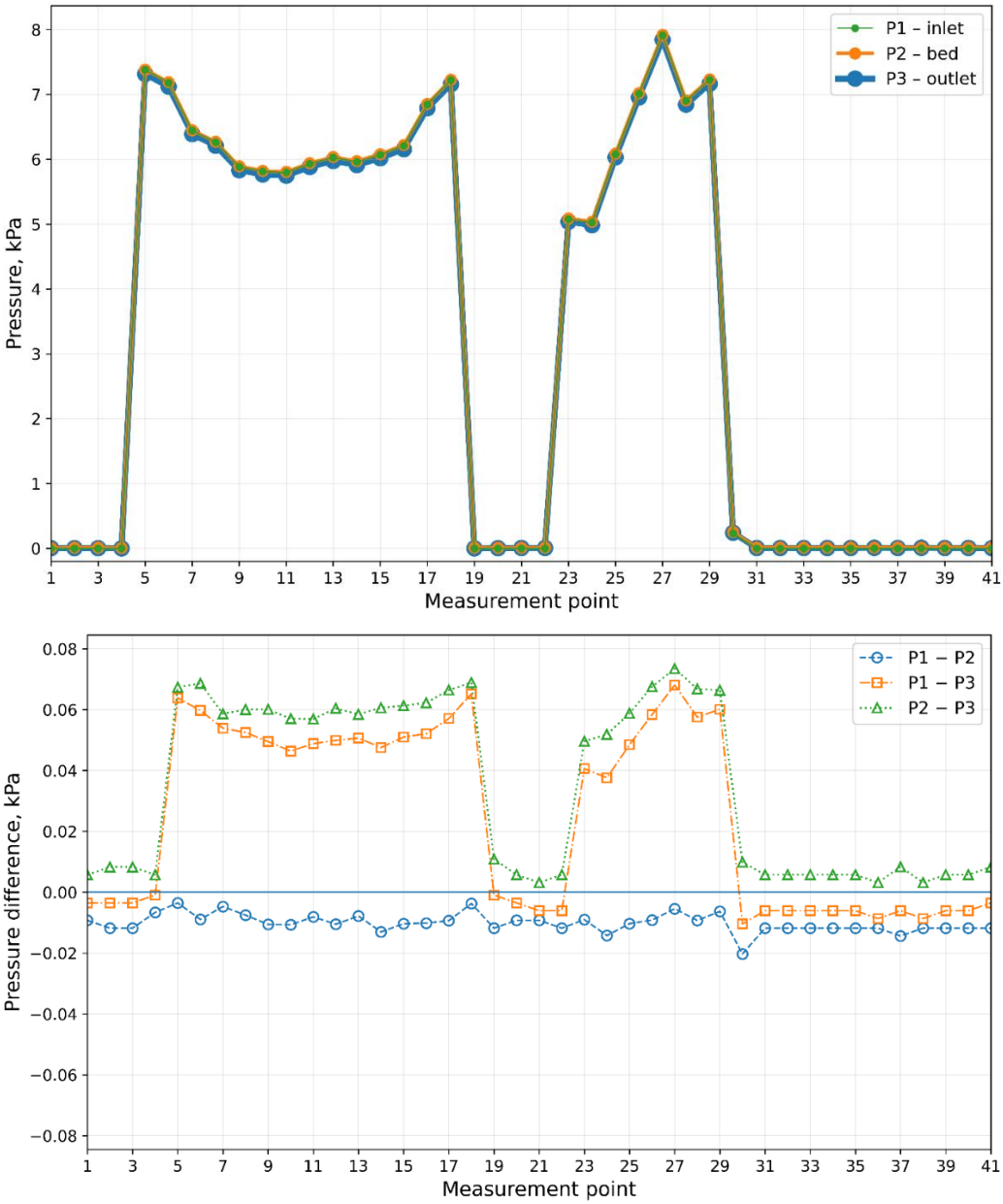


Fig. 3. Inter-channel comparison of pressure measurements under identical applied-pressure conditions: (a) pressure readings of channels P1, P2, and P3; (b) pairwise differences between the pressure-channel readings

curves largely overlapped. The pairwise differences shown in **Fig. 3b** provide a more detailed representation of the inter-channel deviations. The comparison revealed a small but systematic channel-dependent offset, particularly between P3 and the other two channels, supporting the use of individual correction characteristics. The comparison also revealed persistent channel-dependent offsets, particularly between P3 and channels P1 and P2, supporting the use of the individual correction characteristics. Since the verification procedure was based on comparison of the three channels under identical input conditions rather than against an independent reference pressure instrument, the results are interpreted as an assessment of inter-channel consistency rather than absolute measurement accuracy.

The AD7124-4-based temperature-acquisition subsystem was integrated with the Raspberry Pi 5 measurement application, and Pt1000 temperature measurements were tested as part of the commissioning procedure. Functional testing confirmed operation of the multichannel temperature-measurement path, including channel selection through the ADG732 multiplexers, signal conditioning and 24-bit analog-to-digital conversion using the AD7124-4, and data acquisition at the Raspberry Pi level.

The functional tests therefore confirmed operation of both measurement branches and their integration within the common supervisory platform. The pressure subsystem provided corrected and averaged pressure values through the miniSTM32/Modbus RTU acquisition path, while the temperature subsystem provided spatially distributed Pt1000 measurements through the AD7124-4/SPI path. Following commissioning and inter-channel verification, the integrated system was used to acquire pressure and temperature data during the zeolite packed-bed experiments, while the corresponding volumetric gas-flow rate was measured independently using the outlet rotameter.

Gas-flow resistance and thermal response of the zeolite bed

The experimental results obtained for the natural zeolite fraction of 0.5–1.0 mm and a bed height of 0.4 m are presented in **Fig. 4**. Three repeated measurement series were performed using the same eight volumetric oxygen-flow settings. Over the investigated range of $V' = 3.83 \cdot 10^{-9}$ to $3.07 \cdot 10^{-8}$ m³·s⁻¹, the measured pressure drop ranged from 7.93 to 99.79 kPa. Although the overall pressure drop increased with increasing gas-flow rate, the individual measurement series exhibited different degrees of nonlinearity and scatter (**Fig. 4a**). The mean pressure drop calculated from the three series increased from 25.49 kPa at the lowest flow setting to 93.41 kPa at the highest setting. Linear regression of the mean values showed a significant positive relationship between volumetric flow rate and pressure drop $R^2 = 0.840$, $p = 0.0014$. The variability between the individual measurement series was particularly pronounced at the lower flow rates, indicating that the repeatability of the experimental pressure-drop response should be considered when interpreting the gas-flow characteristics of the bed.

The observed overall increase in pressure drop with gas-flow rate is consistent with the expected gas-flow resistance behavior of a granular packed bed. However, the available data characterize the overall flow resistance rather than directly determining the intrinsic permeability of the zeolite bed. A quantitative interpretation using Darcy- or Ergun-type relations would additionally require parameters such as superficial gas velocity, bed porosity, gas properties, and an appropriate flow-regime criterion [8]. Therefore, the present results are interpreted primarily as an experimental characterization of the relationship between gas-flow rate and pressure loss in the investigated zeolite bed.

In contrast to the pressure response, the temperature remained within a very narrow range throughout the experiments (**Fig. 4b**). Across all 24 measurement points, the recorded temperature varied from 16.83 to 16.92 °C, corresponding to a total variation of only 0.09 °C and an overall mean value of approximately 16.88 °C. Identical or nearly identical temperature values were observed at substantially different pressure drops. This

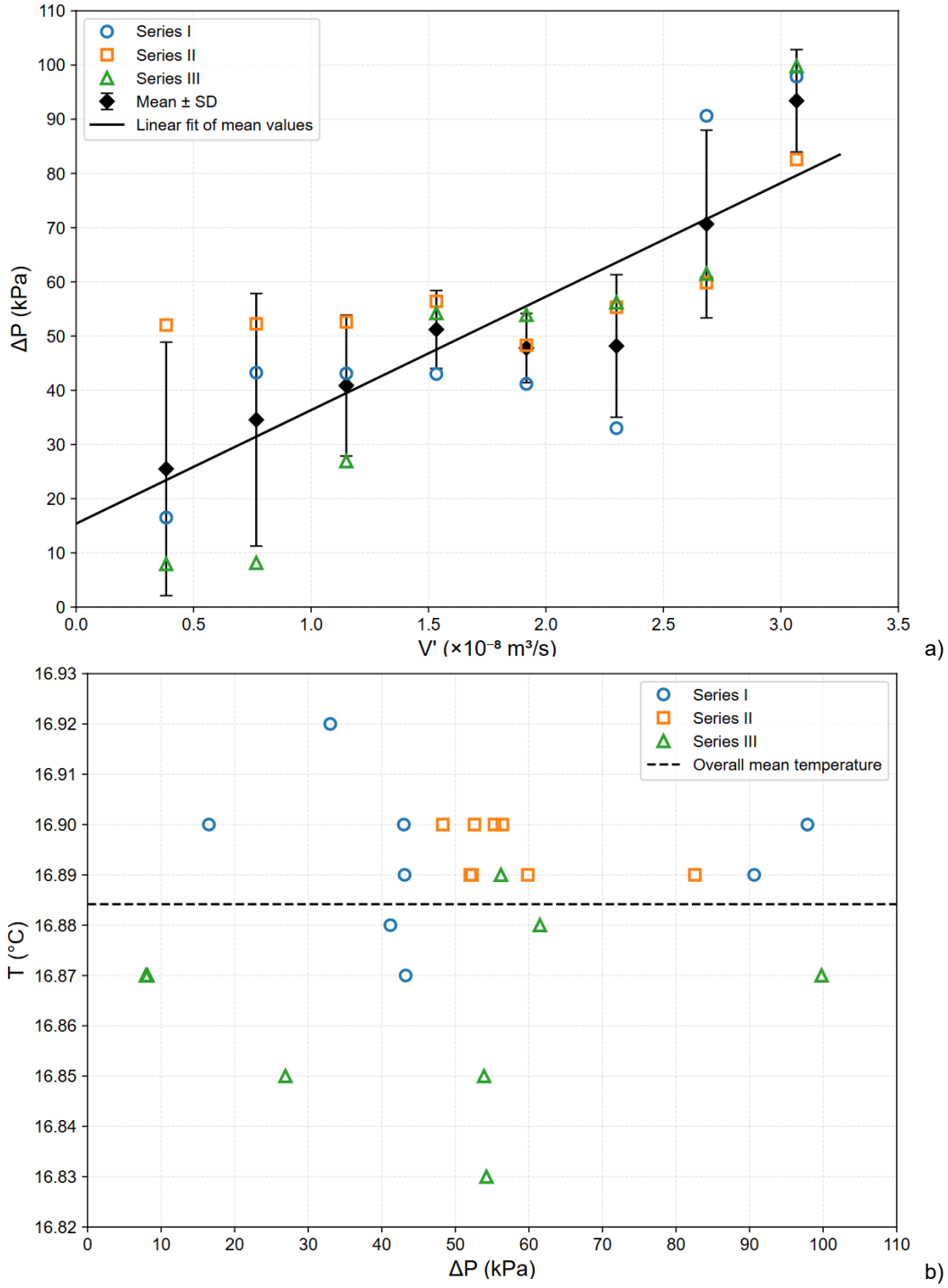


Fig. 4. Experimental characteristics of the natural zeolite packed bed (particle-size fraction 0.5–1.0 mm, bed height 0.4 m): (a) pressure drop versus volumetric oxygen-flow rate, with mean ± SD and linear regression; (b) temperature versus pressure drop, with the overall mean temperature

indicates that changes in the hydraulic resistance of the packed bed were not accompanied by a measurable change in its thermal state under the investigated conditions. Linear regression performed separately for measurement series I, II, and III

resulted in R^2 values of 0.007, 0.180, and 0.004, respectively. In all cases, the relationship was statistically non-significant $p > 0.29$. Analysis of all experimental points together also showed only a negligible correlation between pressure drop and temperature $r = 0.107$, $R^2 = 0.012$, $p = 0.618$. The measured temperature values therefore do not support a monotonic dependence of T on ΔP.

The MATLAB model provided results consistent with the experimentally observed thermal stability of the bed. A comparison of the experimental and MATLAB-calculated mean temperatures for the three measurement series is presented in **Table 3**. The calculated mean temperatures were 16.885, 16.889, and 16.878 °C for measurement series I, II, and III, respectively, compared with the corresponding experimental mean values of 16.891, 16.894, and 16.866 °C. The resulting relative differences did not exceed 0.08%. The model further indicated that the temperature increase associated with viscous dissipation under the investigated flow conditions was below 0.005 °C. This calculated contribution is substantially smaller than the experimentally observed temperature range and supports the conclusion that changes in pressure drop were not accompanied by a detectable systematic thermal response under the investigated conditions.

The observed behavior is also consistent with the broader literature on heat and mass transfer in porous beds. Numerical studies of zeolite-based thermochemical reactors have shown that the thermal response depends strongly on operating conditions, including inlet-gas flow rate and sorbate content [16]. At the particle–pore scale, heat transfer in packed beds is additionally affected by bed porosity, pore structure, and local flow conditions [17]. Although the operating conditions considered in these studies differ from those of the present experiments, their results support the interpretation that the thermal response of a packed bed cannot be inferred from pressure drop alone.

The physical interpretation of this result is that the pressure drop and temperature describe different aspects of the packed-bed behavior. The pressure drop reflects the resistance encountered by the gas flowing through the granular material and therefore varies strongly with the gas-flow conditions. In the present experiments, however, this hydraulic response did not produce a detectable thermal response of the bed, which is also consistent with the numerical results described above. Consequently, repeated temperature values at different pressure drops should not be regarded as anomalous; rather, they indicate that the bed remained approximately thermally stable while its gas-flow resistance changed substantially. Since the observed temperature differences are only of the order of hundredths of a degree Celsius, their interpretation should also take into account the resolution, rounding, and measurement uncertainty of the complete Pt1000 acquisition chain. Until the uncertainty of the complete Pt1000 acquisition chain is quantified, such small differences should not be interpreted as evidence of a separate thermal effect.

CONCLUSIONS

A multichannel embedded measurement system supervised by a Raspberry Pi 5 was developed and functionally evaluated for experimental characterization of a natural zeolite packed bed. The system integrates a miniSTM32-based pressure-measurement

Table 3. **Comparison of experimental and MATLAB-calculated mean temperatures**

Measurement series	ΔP range, kPa	Texp, °C	TMATLAB, °C	Relative difference, %
Series I	16.4–97.9	16.891	16.885	0.035
Series II	48.2–82.6	16.894	16.889	0.029
Series III	8.0–99.8	16.866	16.878	0.071
Overall	8.0–99.8	16.884	16.884	0.000

subsystem with a ten-channel Pt1000 temperature-measurement subsystem based on ADG732 analog multiplexers and an AD7124-4 analog-to-digital converter. Inter-channel verification of the three pressure channels under identical applied-pressure conditions showed close agreement over 41 comparison points, while the observed small channel-dependent deviations supported the use of individual correction characteristics.

Experiments performed with a 0.4 m bed of natural zeolite with a particle-size fraction of 0.5–1.0 mm showed an overall increase in pressure drop with increasing gas-flow rate, although noticeable variability was observed between the three repeated measurement series. The mean pressure drop increased from 25.49 to 93.41 kPa across the investigated gas-flow settings, and linear regression of the mean values confirmed a statistically significant positive relationship R^2 = 0.840, p = 0.0014. The obtained data characterize the overall gas-flow resistance of the investigated packed bed; determination of intrinsic permeability and quantitative interpretation using Darcy- or Ergun-type relations would require additional information on the bed structure and gas-flow conditions.

The measured temperature remained within 16.83–16.92 °C and showed no statistically significant dependence on pressure drop. Thus, substantial changes in gas-flow resistance were not accompanied by a detectable systematic temperature response under the investigated conditions. The developed platform can therefore serve as a basis for further experimental studies of zeolite packed beds, including tests under controlled adsorption–desorption conditions relevant to thermal energy storage. Further development should include metrological characterization of the measurement channels and more detailed analysis of packed-bed flow and thermal behavior.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [G.W., D.A., B.D.]; methodology, [G.W., D.A., P.R.]; software, [V.P., S.S., P.R.]; validation, [G.W., D.A., B.D., P.R.]; formal analysis, [V.P., S.S.]; investigation, [G.W., D.A., B.D.]; data curation, [G.W., D.A., V.P.]; writing – original draft preparation, [G.W., D.A., B.D., V.P.]; writing – review and editing, [V.P., S.S., P.R.]; visualization, [V.P.]; supervision, [G.W., S.S.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Feng, C., E, J., Han, W., Deng, Y., Zhang, B., Zhao, X., & Han, D. (2021). Key technology and application analysis of zeolite adsorption for energy storage and heat-mass transfer process: A review. Renewable and Sustainable Energy Reviews, 144, 110954. https://doi.org/10.1016/j.rser.2021.110954

[2] de Gennaro, B., Cappi, A., de Gennaro, M., Bianco, N., Langella, A., Cappelletti, P., Marocco, A., Aprea, P., & Pansini, M. (2022). Use of zeolites in the capture and storage of thermal energy by water desorption–adsorption cycles. Materials, 15(16), 5574. https://doi.org/10.3390/ma15165574

[3] Kouchachvili, L., Bardy, D. A., Djebbar, R., & Hogg, L. E. W. (2023). Natural zeolites as host matrices for the development of low-cost and stable thermochemical energy storage materials. Journal of Porous Materials, 30, 163–173. https://doi.org/10.1007/s10934-022-01277-3

[4] Abedin, A. H., & Rosen, M. A. (2012). Closed and open thermochemical energy storage: Energy- and exergy-based comparisons. Energy, 41(1), 83–92. https://doi.org/10.1016/j.energy.2011.06.034
[5] Yu, N., Wang, R. Z., & Wang, L. W. (2013). Sorption thermal storage for solar energy. Progress in Energy and Combustion Science, 39(5), 489–514. https://doi.org/10.1016/j.pecs.2013.05.004
[6] Gao, S., Wang, S., Hu, P., Wang, J., Sun, Y., & Ma, Z. (2023). Performance of sorption thermal energy storage in zeolite bed reactors: Analytical solution and experiment. Journal of Energy Storage, 64, 107154. https://doi.org/10.1016/j.est.2023.107154
[7] Kim, H., Hong, S. K., Kang, J. G., Moon, S. W., Kim, G. H., Yoon, S., Wei, D. J., Moon, H., & Kim, B. S. (2024). Exploration adsorption characteristics of zeolite 13X depending on humidity and flow rate in sorption thermal energy storage applications. International Journal of Heat and Mass Transfer, 221, 125049. https://doi.org/10.1016/j.ijheatmasstransfer.2023.125049
[8] Hassan, E. B. E., & Hoffmann, J. (2024). Review on pressure drop through a randomly packed bed of crushed rocks. Discover Applied Sciences, 6, 126. https://doi.org/10.1007/s42452-024-05773-w
[9] Ahmed, Z., Constantin, A., & Bindra, H. (2022). The thermal response of a packed bed thermal energy storage system upon saturated steam injection using distributed temperature sensing. Energies, 15(10), 3704. https://doi.org/10.3390/en15103704
[10] Wang, T., Wang, Y., Zhang, Y., & Chen, Y. (2026). Reaction zone evolution governing thermal output in a zeolite 13X sorption reactor: An experimental study. Energies, 19(8), 1874. https://doi.org/10.3390/en19081874
[11] Knapp, A., & Bloom, A. J. (2022). Easy as piadcs: A low-cost, ultra-high-resolution data acquisition system using a Raspberry Pi. Applications in Plant Sciences, 10(3), e11485. https://doi.org/10.1002/aps3.11485
[12] Véliz, M. E., Real, G. E., & Otero, A. D. (2024). Flexible and configurable embedded electrical energy measurement system to acquire and process high-frequency features. HardwareX, 19, e00539. https://doi.org/10.1016/j.ohx.2024.e00539
[13] Evangelisti, L., De Cristo, E., Monteleone, S., Guattari, C., Gori, P., Pini, I., de Rubeis, T., & Ambrosini, D. (2025). A new customized measurement system for a non-contact, enhanced thermometric method. Energies, 18(6), 1537. https://doi.org/10.3390/en18061537
[14] Amorim, M. L. M., Ando Junior, O. H., Gazziro, M., & Carmo, J. P. P. (2026). Nexus: A modular open-source multichannel data logger – Architecture and proof of concept. Automation, 7(1), 25. https://doi.org/10.3390/automation7010025
[15] Payet, P., Fakra, D. A. H., Rakotosaona, R., Ratsimba, M. H., & Randrianarison, M. P. (2025). Design of a highly affordable and reliable data acquisition unit system dedicated to experimental research and education. Measurement, 251, 117187. https://doi.org/10.1016/j.measurement.2025.117187
[16] Abohamzeh, E., & Frey, G. (2022). Numerical investigation of the adsorption process of zeolite/water in a thermochemical reactor for seasonal heat storage. Energies, 15(16), 5944. https://doi.org/10.3390/en15165944
[17] Zou, Y., Kuang, S., Wu, Y., & Yu, A. (2025). Particle-pore-scale modelling and analysis of heat transfer in packed beds: Effects of bed porosity and pore structure. Applied Thermal Engineering, 279, 127601. https://doi.org/10.1016/j.applthermaleng.2025.127601

БАГАТОКАНАЛЬНА ВИМІРЮВАЛЬНА СИСТЕМА НА БАЗІ RASPBERRY PI 5 ДЛЯ ДОСЛІДЖЕННЯ ОПОРУ ГАЗОВОМУ ПОТОКУ ТА МОНІТОРИНГУ ТЕМПЕРАТУРИ В ШАРІ ПРИРОДНОГО ЦЕОЛІТУ

Ґжегож Вальовські[1], Дорота Андерс[1], Барбара Дибек[1], Вадим Пташник[2], Сергій Сиротюк[2]*, Патрик Радек[3]
[1]Інститут технологічних та природничих наук — Національний дослідний інститут, вул. Грабська, 3, Фаленти, 05-090 Рашин, ПОЛЬЩА
[2]Факультет механіки, енергетики та інформаційних технологій, Львівський національний університет ветеринарної медицини та біотехнологій імені С. З. Ґжицького, вул. Володимира Великого, 1, м. Дубляни, 79083, УКРАЇНА
[3]Кафедра оптоелектроніки, Факультет електротехніки, Сілезька політехніка, вул. Болеслава Кривоустого, 2, 44-100 м. Глівіце, ПОЛЬЩА

АНОТАЦІЯ

Вступ. Експериментальне дослідження гранульованих шарів для сорбційного акумулювання теплової енергії потребує одночасного вимірювання опору газовому потоку та температури із застосуванням гнучкої багатоканальної системи збору даних. У роботі розроблено вимірювальну систему під керуванням Raspberry Pi 5 для дослідження шару природного кліноптилоліту.

Матеріали та методи. Експерименти проводили в теплоізольованому реакторі об'ємом 0,5 м³, що містив шар кліноптилоліту висотою 0,4 м із фракцією частинок 0,5–1,0 мм. Робочим газом був кисень. Сигнали тиску оброблялися підсистемою на базі miniSTM32, тоді як для моніторингу температури використовували десять датчиків Pt1000, мультиплексори ADG732 та перетворювач AD7124-4. Було виконано три повторні серії вимірювань за восьми значень витрати газу. Дані залежності тиску від витрати аналізували з використанням середніх значень, вибіркового стандартного відхилення та лінійної регресії; залежність температури від перепаду тиску оцінювали за допомогою лінійної регресії та коефіцієнта кореляції Пірсона. Для оцінювання температурної реакції використовували стаціонарну двовимірну модель, реалізовану в MATLAB.

Результати. Функціональну працездатність платформи було перевірено, зокрема шляхом порівняння трьох каналів вимірювання тиску в 41 контрольній точці, яке показало добру міжканальну узгодженість. Перепад тиску змінювався від 7,93 до 99,79 кПа, тоді як його середнє значення зросло від 25,49 до 93,41 кПа. Лінійна регресія показала статистично значущий позитивний зв'язок між витратою газу та перепадом тиску (R^2 = 0,840, p = 0,0014). Температура залишалася в межах 16,83–16,92 °C, при цьому статистично значущої залежності від перепаду тиску не виявлено. Середні значення температури, розраховані в MATLAB, відрізнялися від експериментальних не більше ніж на 0,08 %, а розраховане підвищення температури внаслідок в'язкої дисипації становило менше 0,005 °C.

Висновки. Розроблена система забезпечує інтегроване вимірювання опору газовому потоку та температури в шарі природного кліноптилоліту. Експериментальні та чисельні результати свідчать, що за досліджених умов суттєві зміни опору газовому потоку не супроводжувалися виявленою систематичною температурною реакцією. Запропонована архітектура може бути основою для подальших досліджень цеолітових шарів за контрольованих сорбційних умов, релевантних для систем акумулювання теплової енергії.

Ключові слова: вбудована вимірювальна система, Raspberry Pi 5, збір даних, опір газовому потоку, природний цеоліт, зернистий шар, акумулювання теплової енергії.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
30 April, 2026	08 June, 2026	10 June, 2026	28 September, 2026

Electronics and Information Technologies, 2026, 35, 219–230
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 535.37; 537.311

SPECTRAL AND ELECTRICAL CHARACTERISTICS OF Mn- AND Mn-Zr-DOPED $BaGa_2O_4$ PHOSPHORS

Dmytro Slobodzyan**, Markiyan Kushlyk, Vyacheslav Vasyltsiv, Ivanna Medvid, Andriy Luchechko
Department of Sensor and Semiconductor Electronics,
Ivan Franko National University of Lviv
107 Gen. Tarnavskoho Street, Lviv, 79017, UKRAINE

Slobodzyan, D., Kushlyk, M., Vasyltsiv, V., Medvid, I., Luchechko, A. (2026). Spectral and Electrical Characteristics of Mn- and Mn-Zr-Doped $BaGa_2O_4$ Phosphors. *Electronics and Information Technologies*, 35, 219–230. https://doi.org/10.30970/eli.35.16

ABSTRACT

Background. $BaGa_2O_4$ is a wide-band-gap oxide with promising luminescent and electronic properties, making it a potential host for phosphor and optoelectronic materials. Controlled doping provides an effective approach for modifying its luminescence, defect structure, and charge-transport properties.

Materials and Methods. Undoped $BaGa_2O_4$ (BGO), BGO: 0.1 mol% Mn, and BGO: 0.1 mol% Mn, 0.2 mol% Zr were synthesized by the high-temperature solid-phase method. Photoluminescence and photoluminescence excitation spectra were measured at room temperature. DC conductivity was investigated in vacuum in the temperature range of 370–490 K, while AC conductivity was measured in the frequency range of 100 Hz–50 kHz using an excitation electrical field amplitude of 5.0 V.

Results and Discussion. Undoped BGO exhibits intense broadband blue luminescence with a maximum near 410 nm, attributed to host emission centers. Mn doping shifts the emission maximum to 430-440 nm, significantly decreases the intensity, and introduces an additional excitation band at 270–340 nm, indicating the existence of an effective excitation channel associated with Mn centers. Co-doping with Zr further decreases the intensity but does not substantially change the emission spectral composition. According to the CIE 1931 analysis, the color purity decreases from 80% for BGO to 75% and 71% for Mn- and Mn-Zr-doped samples, respectively. Electrical measurements show that Mn doping increases DC conductivity by more than an order of magnitude, whereas Mn-Zr co-doping reduces the conductivity while maintaining values above those of undoped BGO. The DC activation energies remain close, ranging from 1.15 to 1.19 eV. AC conductivity exhibits frequency dispersion described by the universal Jonscher law, with the temperature dependence of the exponent *s* indicating changes in localized charge-transport mechanisms.

Conclusion. Mn and Mn-Zr doping effectively modify the optical and electrical properties of barium gallate through changes in localized states and defect structure. The ability to tune luminescence and charge transport highlights the potential of doped $BaGa_2O_4$ for phosphor and optoelectronic applications.

Keywords: $BaGa_2O_4$, Mn-Zr dopants, luminescence, electrical conductivity, localized states, phosphors

© 2026 Dmytro Slobodzyan et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

$BaGa_2O_4$ is a wide-band-gap oxide with a hexagonal structure formed by a framework of GaO_4 tetrahedra and containing channels occupied by Ba^{2+} ions [1]. Owing to its high thermal and chemical stability, structural robustness, and ability to accommodate optically active impurities, $BaGa_2O_4$ is considered a promising host matrix for luminescent and multifunctional oxide materials [2]. Modification of its optical and electronic properties through controlled doping makes this material attractive not only for phosphor applications but also for emerging optoelectronic technologies, where the combination of light emission, charge transport, and defect-state engineering is particularly important.

The optical and electrical properties of wide-band-gap oxides are strongly governed by their defect structure, including intrinsic point defects, impurity centers, charge-compensating defects, and localized electronic states [3]. These states can influence carrier generation, trapping, transport, and radiative recombination, establishing a direct relationship between the luminescent and electrical characteristics of the material. Doping with transition-metal ions, such as Mn, provides an effective approach for introducing additional localized states and modifying the concentration and distribution of defect centers, thereby enabling control over both the emission characteristics and charge-transport processes [4]. Such defect engineering is particularly relevant for optoelectronic materials, since the performance of luminescent and light-sensing devices strongly depends on the balance among carrier localization, charge transport, and radiative recombination.

Co-doping with ions of different valence states and ionic radii provides an additional means of tailoring the defect structure and local charge compensation in the $BaGa_2O_4$ lattice. In particular, the simultaneous incorporation of Mn and Zr may modify the concentration and spatial distribution of localized states and intrinsic point defects, thereby affecting the pathways of charge transfer and the magnitude of electrical conductivity. Changes in these processes can, in turn, influence the interaction between charge carriers and luminescent centers. Similar relationships between doping-induced defect modification, charge transport, and optical response have been observed in other wide-band-gap gallium-containing oxides [5].

From the viewpoint of optoelectronic applications, simultaneous consideration of luminescence and electrical properties is especially important. A material intended for light-emitting, photodetection, scintillation, or other optoelectronic functions should possess not only suitable emission characteristics but also controlled charge-carrier transport and trapping behavior. Therefore, investigation of both DC and AC conductivity can provide complementary information about the defect states and localized charge-transfer mechanisms responsible for the functional properties of doped $BaGa_2O_4$. In particular, the activation energy obtained from the temperature dependence of DC conductivity provides information about the energy barrier governing the dominant transport process, whereas the frequency dependence of AC conductivity can reveal the contribution of localized carriers and their interaction with the surrounding defect environment.

Although the structural and luminescent properties of $BaGa_2O_4$ and its doped derivatives have been investigated [1, 2, 6], the relationship between impurity incorporation, luminescence, charge transport, and localized-state dynamics remains insufficiently understood. In particular, systematic studies combining the spectral characteristics with the temperature and frequency dependences of DC and AC conductivity in Mn-doped and Mn-Zr-co-doped $BaGa_2O_4$ are limited. Analysis of these characteristics is important for understanding how doping affects the localized charge-transfer processes, and for assessing the potential of these materials as functional components in optoelectronic applications.

Therefore, the aim of this work is to investigate the spectral and electrical characteristics of Mn- and Mn-Zr-doped $BaGa_2O_4$ phosphors and to elucidate the influence of doping on their spectral response and charge-transport behavior. The combined analysis of spectral and electrical properties provides insight into the role of doping-induced defects and localized states and helps evaluate the prospects of Mn- and Mn-Zr-doped $BaGa_2O_4$ for phosphor and optoelectronic applications.

MATERIALS AND METHODS

The synthesis of the studied $BaGa_2O_4$ samples was carried out using a high-temperature solid-phase method. Powders of barium carbonate ($BaCO_3$), gallium oxide (Ga_2O_3), manganese (III) oxide (Mn_2O_3), and zirconium (IV) oxide (ZrO_2) with a purity of at least 99.99% were used as initial precursors. The calculation of impurity concentrations for doping was performed using the isomorphic substitution model for Ga^{3+} ions in the cationic sublattice [6]. The weighed oxide mixtures were ground in an agate mortar with the addition of isopropyl alcohol for 6 h. After that, the powder mixtures were formed into disk-shaped samples with a diameter of 6 mm and a thickness of ~1.5 mm under a pressure of 100 atm, and then sintered at 1523 K for 12 h. Three types of samples were prepared: undoped base matrix (BGO), samples with 0.1 mol% Mn (BGO:Mn), and samples doped with 0.1 mol% Mn and 0.2 mol% Zr (BGO: Mn, Zr). According to the results of X-ray diffraction analysis, all samples contained a single $BaGa_2O_4$ phase.

Photoluminescence (PL) and photoluminescence excitation (PLE) spectra were recorded at room temperature using a Solar CM2203 spectrofluorometer. The PL spectra were automatically corrected for the spectral response, and the PLE spectra for the Xenon lamp emission profile.

To study the electrophysical properties of the samples, a highly conductive silver paste was applied to the opposite surfaces of each disk. The diameter of the formed metal contacts was 5 mm, ensuring a uniform electric-field distribution throughout the volume of the material under study. DC conductivity measurements were performed in vacuum during cooling from approximately 490 to 370 K. The frequency dependence of AC conductivity was studied using a HIOKI IM3536 LCR meter over the range 100 Hz–50 kHz at an excitation electrical field amplitude of 5.0 V.

RESULTS AND DISCUSSION

In the luminescence spectrum of undoped BGO samples when excited at a wavelength of 250 nm, i.e., in the region of the fundamental absorption edge, an intense broad emission band with a maximum near 410 nm is observed, which covers the spectral range from approximately 325 to 625 nm (**Fig. 1a, curve 1**). The broad nature of this band indicates its non-elementary nature and the possible superposition of several closely spaced luminescence bands associated with different emission centers. The observed luminescence in the blue-violet region of the spectrum can be attributed to intrinsic defects or localized electronic states in the BGO matrix.

After the introduction of a manganese impurity, the luminescence intensity significantly decreases compared to undoped samples (**Fig. 1a, curve 2**). At the same time, the broadband maximum shifts to the long-wavelength region, approximately 430 nm. Such a change in the spectrum indicates that the introduction of Mn significantly affects the relaxation processes of the electronic excitation energy in the matrix. The decrease in the intensity of intrinsic luminescence may be associated with the interception of excitation energy by Mn centers and subsequent relaxation via the states of these impurity ions. In this case, the shift of the maximum to the long-wavelength region may indicate a decrease in the contribution of the short-wavelength component of intrinsic luminescence and a simultaneous increase in the relative contribution of the radiation associated with Mn^{2+}.

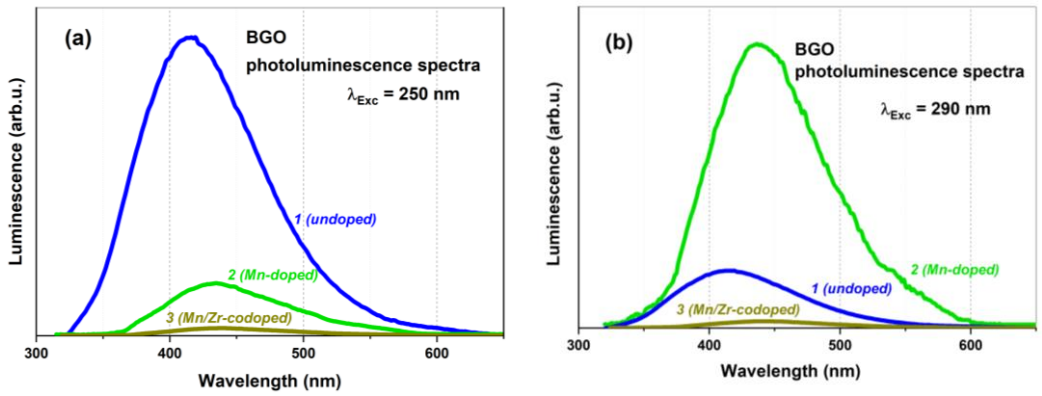


Fig. 1. Luminescence spectra of undoped (curve 1), doped with Mn impurity (curve 2), and co-doped with Mn-Zr impurities (curve 3) BGO samples when excited in the region of the fundamental absorption edge at 250 nm (a) and when excited at a wavelength of 290 nm (b)

Mn^{2+} ions in the crystal field are characterized by broad luminescence bands associated mainly with intracenter *d–d* transitions. Their spectral position and intensity depend significantly on the coordination environment of Mn^{2+}, the symmetry of the local crystal field, and the nature of the interaction between the impurity center and the matrix. Therefore, the observed long-wavelength emission in the region of 430–440 nm can be attributed to Mn^{2+} centers, although additional data are required to unambiguously determine the specific transition scheme, in particular, temperature dependence and time-resolved luminescence measurements.

Co-doping with manganese and zirconium (Mn-Zr) leads to a further decrease in the luminescence intensity (**Fig. 1a, curve 3**). This indicates that the introduction of Zr additionally changes the relaxation mechanism of electronic excitation in BGO. Most likely, this is due to changes in the concentration and nature of defect states, in the local crystal field near the Mn^{2+} centers, or to the appearance of additional non-radiative relaxation channels [7].

When excited at a wavelength of 290 nm, which corresponds to the region of much weaker absorption of the BGO matrix, the spectral pattern changes significantly (**Fig. 1b**). For undoped samples, the luminescence intensity under such excitation decreases sharply, which indicates a low efficiency of excitation of the intrinsic emission centers outside the region of the fundamental absorption edge. In contrast, in BGO:Mn samples excited at 290 nm, a broad luminescence band with a maximum near 440 nm forms. Its intensity is several times higher, and its spectral half-width is somewhat narrower than that of undoped BGO samples. This is an important argument that, at 290 nm, excitation is more selectively directed to impurity Mn centers, whereas at 250 nm, both the matrix's own centers and the impurity centers associated with manganese ions are simultaneously excited.

Thus, changing the excitation wavelength allows for partial separation of the matrix and Mn impurity contributions to the luminescence spectrum. Excitation at 250 nm preferentially activates BGO intrinsic centers through absorption in the region of the fundamental absorption edge, while excitation at 290 nm more effectively activates Mn-related impurity centers.

The luminescence excitation spectra presented in **Fig. 2** further confirm the difference in the luminescence mechanisms in undoped and doped samples. For undoped BGO, when recording radiation at 410 nm (**Fig. 2a, curve 1**), the main excitation band is located in the region of the fundamental absorption edge of the matrix. At the same time, in the region of BGO transparency, the luminescence excitation efficiency is low. After Mn doping, the excitation spectrum changes significantly; in particular, alongside the weak components associated with matrix absorption, a broad band appears in the range of approximately

260 – 340 nm (**Fig. 2a, curve 2**). Its presence indicates the appearance of an additional absorption channel associated with impurity Mn centers. Accordingly, Mn luminescence can be effectively excited without direct excitation of the interband transitions of the matrix.

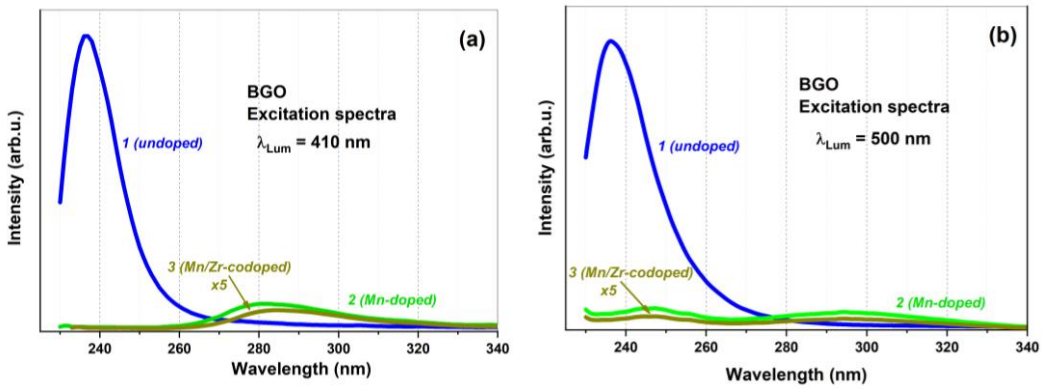


Fig. 2. Luminescence excitation spectra of undoped (curve 1), doped with Mn impurity (curve 2), and co-doped with Mn-Zr impurities (curve 3) BGO samples when recording the luminescence at 410 nm (a) and 500 nm (b)

When recording luminescence at 500 nm (**Fig. 2b**), the contribution of impurity centers is even more pronounced. At the same time, co-doping with Zr changes the excitation spectra (**Fig. 2b, curve 3**), indicating the influence of Zr on the formation and relaxation of excited states. The decrease in luminescence intensity in Mn-Zr samples may be associated with reduced radiative relaxation efficiency or increased non-radiative relaxation [7]. Thus, the results of excitation spectroscopy confirm the existence of an excitation channel of Mn centers in BGO and explain the increase in the relative contribution of long-wavelength luminescence when excited at 290 nm.

The additional UV excitation band with a maximum near 280 – 290 nm in the Mn-doped samples may be associated with a charge-transfer transition from O^{2-} ligands to Mn^{2+} ions ($O^{2-} \rightarrow Mn^{2+}$), similar to charge-transfer excitation bands reported for Mn-doped oxide materials, including zinc and magnesium gallates. Thus, the appearance of this band indicates the formation of an additional excitation pathway involving Mn^{2+} centers. Consequently, Mn-related luminescence can be efficiently excited not only through excitation of the BGO host matrix but also through direct charge-transfer excitation of the Mn^{2+} centers.

The color characteristics of the studied BGO samples were evaluated using the CIE 1931 chromaticity diagram (**Fig. 3**). Undoped BGO is characterized by chromaticity coordinates located in the blue region of the diagram, and the calculated color purity is approximately 80%. For Mn-doped samples, a moderate shift of the chromaticity coordinates towards the green region and a decrease in color purity to approximately 75% are observed. This change is consistent with the shift of the luminescence spectrum to the long-wavelength region after the introduction of Mn and an increase in the contribution of broadband radiation from Mn centers. Co-doping of Mn and Zr leads to a further change in the chromaticity coordinates and a decrease in the calculated color purity to approximately 71%. At the same time, the change in the coordinates is relatively small, indicating Zr mainly affects the luminescence intensity and the efficiency of relaxation processes, and does not radically change the spectral composition of the emission.

Thus, the CIE 1931 results show that the introduction of Mn and subsequent co-doping with Zr do not cause a fundamental change in the color of the BGO emission, but gradually shift it from the blue region towards the greenish-blue and are accompanied by a decrease

in color purity. The main effect of doping is manifested not so much in the change in the chromaticity coordinates, but in a significant change in the intensity and ratio of the intrinsic and impurity components of luminescence.

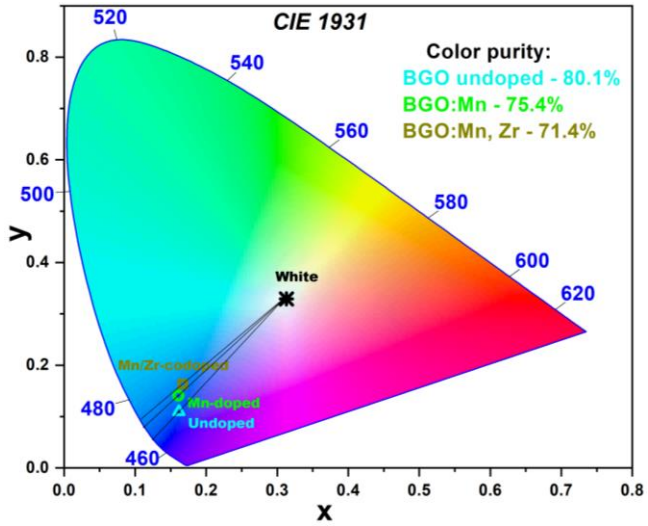


Fig. 3. CIE 1931 chromaticity diagrams of undoped, Mn-doped, and Mn-Zr-codoped BGO samples. The color-purity values are also indicated for each sample

Overall, the results indicate that Mn is an active center that significantly alters the mechanism of excitation-energy relaxation in BGO, and that the introduction of Zr further modifies these processes. In this case, excitation at 290 nm is particularly effective for selectively exciting Mn centers, as evidenced by the formation of an intense, broad luminescence band with a maximum at about 440 nm.

Fig. 4 shows the temperature dependence of the DC conductivity of the studied samples. For all samples, an increase in conductivity with temperature is observed, indicating that the dominant transport process is thermally activated, a characteristic of wide-gap oxide semiconductors and dielectrics. Undoped BGO samples demonstrate the lowest conductivity values (σ_{DC}), while doping with manganese ions leads to an increase

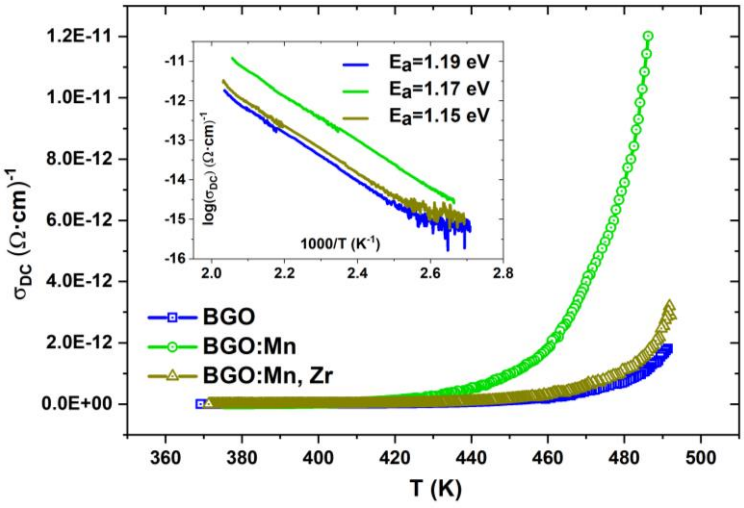


Fig. 4. Temperature dependences of the DC conductivity of BGO, BGO:Mn, and BGO:Mn, Zr samples

of more than an order of magnitude in the entire studied temperature range. For co-doped BGO:Mn, Zr samples, the conductivity values occupy an intermediate position between undoped and Mn-doped samples.

In Arrhenius coordinates, the temperature dependences are close to linear in the range of conductivity changes by about four orders of magnitude. The calculated activation energy values are 1.19 eV for BGO, 1.17 eV for BGO:Mn, and 1.15 eV for BGO:Mn, Zr. The close agreement among the activation energy values indicates that Mn doping and Mn/Zr co-doping do not significantly change the energy barrier of the dominant transport process. This is also consistent with the behavior of deep centers in related oxides [8]. Given the wide-bandgap nature of BGO and its high activation energy, the observed conductivity can be attributed to the thermally activated release of charge carriers from localized defect states. One possible type of such states is deep electronic centers formed by oxygen vacancies.

The increase in DC conductivity upon Mn doping indicates that the introduction of manganese changes the electrical properties of BGO, although the obtained activation energy values remain close to those for the undoped matrix. Unlike gallium oxide [7], manganese ions do not create additional fast recombination channels in BGO. One possible explanation for the observed increase in conductivity may be the formation of localized electronic states associated with Mn centers or a change in the equilibrium of intrinsic defects. In particular, during high-temperature solid-state synthesis, some manganese ions may transition to the reduced state, Mn^{2+}. In the case of substitution of trivalent gallium, such an ion acts as an effective negatively charged acceptor center. Its compensation may occur, in particular, due to the formation of oxygen vacancies. In this case, a change in the concentration of oxygen vacancies can affect the concentration of charge carriers and the electrical conductivity of the material [9]. A similar effect of transition metal impurities on electrical conductivity was previously observed in related wide-gap gallium-containing oxides [10].

Co-doping with zirconium reduces conductivity relative to samples doped with Mn alone. This behavior is consistent with the assumption that Zr modifies the defect equilibrium in the material and partially compensates for defects that contribute to charge transfer in Mn-containing samples [11]. The fact that the conductivity of the co-doped samples remains higher than that of the undoped matrix may indicate incomplete charge compensation or the formation of complex local defect clusters composed of manganese and zirconium ions, which can participate in charge-transfer processes, thereby reducing the overall resistance of the material [12]. Thus, the obtained data indicate a significant effect of doping on the absolute conductivity but do not allow us to unambiguously establish the charge-transfer mechanism. In addition to electronic conductivity, BGO may have quite significant ionic conductivity through channels present in the material structure.

AC conductivity studies (**Fig. 5**), conducted over the range 100 Hz to 50 kHz, reveal a significant frequency dispersion, which is typical of dielectric materials and wide-gap semiconductors with a high density of localized states. At 303 K, the conductivity of all samples increases monotonically with increasing applied electric-field frequency. The absolute values of AC conductivity at this temperature lie in the range from ~ 5×10^{-12} to ~ 5×10^{-10} $(\Omega\cdot cm)^{-1}$. The highest conductivity values are demonstrated by the Mn-doped sample, while the undoped matrix is characterized by the lowest values. With a further increase in frequency to 50 kHz, the curves partially converge, which may indicate a decrease in the contribution of long-range charge transfer and an increase in the role of localized relaxation and polarization processes [13].

When the temperature increases to 393 K, the overall AC conductivity increases, reaching values in the range of ~ 5×10^{-11} to 10^{-9} $(\Omega\cdot cm)^{-1}$. At this temperature, in the frequency region, approximately from 100 Hz to 1 kHz, there is a tendency for the curves

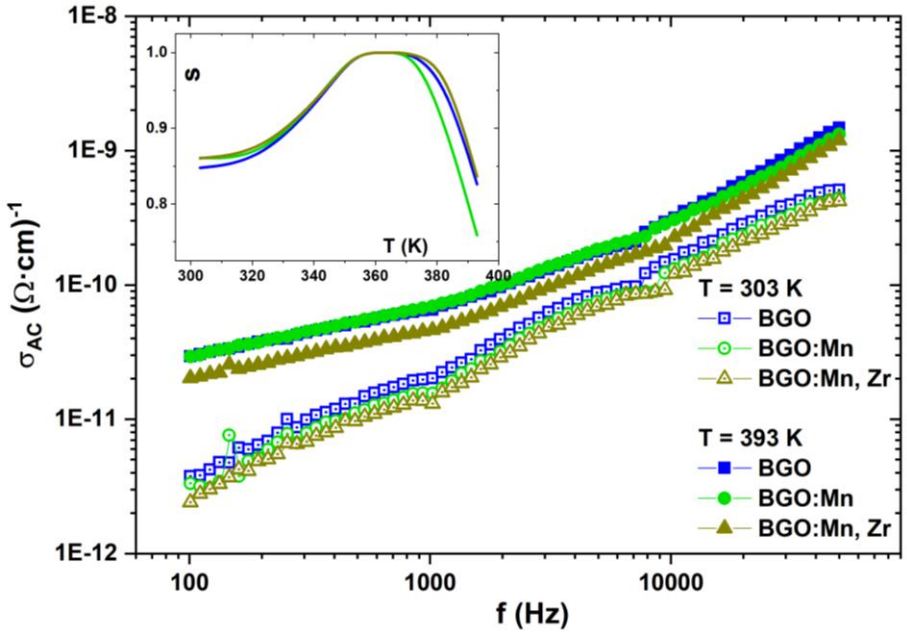


Fig. 5. Frequency dependences of AC conductivity of BGO, BGO:Mn, and BGO:Mn, Zr samples at different temperatures

to plateau. The presence of such a plateau may be associated with the dominance of the DC component of conductivity at low frequencies.

In the high-frequency region, the AC conductivity exhibits a dispersion characteristic of disordered and defective oxide systems and can be described by the universal Johnsher law [14]:

$$\sigma_{AC}(\omega) = \sigma_{DC} + A \cdot \omega^{s}, \quad (1)$$

where σ_{AC} is the total AC conductivity, σ_{DC} is the DC component of conductivity, A is the temperature-dependent coefficient, ω is the angular frequency, and s is a dimensionless frequency dependence index that characterizes the nature of the frequency dispersion and is associated with the mechanism of localized charge transfer.

The temperature dependence of the parameter s has a non-monotonic character. At 303 K, its values are approximately 0.85 – 0.86, after which they increase to values close to unity in the region of 360 – 370 K, and decrease with further increases in temperature.

The increase of the parameters to values close to unity may indicate a change in the nature of the local relaxation of charge carriers. Such behavior can be considered within the framework of models of tunneling of carriers between localized states, in particular, the non-overlapping small polaron tunneling (NSPT) model [13], as well as the nearly constant loss (NCL) model [15]. At the same time, the obtained results do not allow us to unambiguously give preference to one of these models, and therefore, both interpretations should be considered as possible.

With a further increase in temperature above ~ 370 K, a decrease in the parameter *s* is observed. Such an inversion of the temperature dependence may indicate a change in the dominant charge transfer mechanism and is characteristic of the correlated barrier hopping (CBH) model, within which the *s* index decreases with increasing temperature [13]. This may be due to thermally activated charge transfer between localized states through potential barriers. With increasing temperature, the probability of thermally activated jumps increases, which leads to a change in the frequency dependence of the conductivity [16].

For Mn-doped samples, the transition to the region of decreasing *s* is observed already at approximately 365 K, and the minimum values of the parameter reach approximately 0.75. This may indicate that the introduction of Mn changes the local environment of the charge carrier localization centers and affects the nature of their frequency-dependent transport. Substitution of Ga by Mn ions with a different ionic radius can cause local structural deformations and change the distribution of potential barriers between neighboring localization centers. Thus, even at close values of the activation energy of DC conduction, doping can significantly affect the local dynamics of charge carriers.

CONCLUSION

It was found that undoped $BaGa_2O_4$ exhibits intense broadband luminescence with a maximum near 410 nm, attributed to intrinsic emission centers in the matrix. Mn doping is accompanied by a shift of the luminescence maximum to 430 nm and a significant decrease in its intensity when excited in the region of the fundamental absorption edge at a wavelength of 250 nm. When excited at 290 nm, a broad luminescence band with a maximum near 440 nm is formed in $BaGa_2O_4$ samples. The appearance of an excitation band in the range of 270–340 nm indicates the presence of an effective excitation channel associated with Mn centers and the possibility of their more selective excitation compared to the intrinsic centers of the matrix. Co-doping of Mn-Zr leads to a further decrease in luminescence intensity, which may be due to changes in the matrix defect structure and an enhancement of non-radiative relaxation. At the same time, the introduction of Zr does not significantly change the spectral composition of the emission.

According to the CIE 1931 chromaticity diagrams, undoped $BaGa_2O_4$ exhibits blue emission with a color purity of about 80%. Mn doping and Mn/Zr co-doping cause a moderate shift of the chromaticity coordinates towards the green region and a decrease in color purity to about 75 and 71%, respectively. The results demonstrate the possibility of controlling the spectral and color characteristics of $BaGa_2O_4$ luminescence through doping and by varying the excitation wavelength, which opens prospects for its use in optoelectronic applications.

Doping BaGa2O4 with Mn impurities increases DC conductivity relative to the undoped matrix. Co-doping with Zr/Mn impurities reduces conductivity compared to samples doped only with Mn, however, its values remain higher than for the undoped matrix. The activation energies of DC conductivity for the studied samples are close, at 1.15 – 1.19 eV. This indicates that doping does not significantly change the energy barrier of the dominant transport process. The observed change in the absolute conductivity may be associated with a modification of the concentration of localized states and intrinsic point defects.

The frequency dependence of the AC conductivity is described by the universal Johnsher law. The non-monotonic temperature dependence of the parameter *s* indicates a change in the nature of localized charge transfer as temperature varies. The increase in *s* to values close to unity can be associated with tunneling or almost frequency-independent transfer mechanisms, while the further decrease in *s* with increasing temperature is compatible with the correlated barrier jump model.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by the National Research Foundation of Ukraine, Grant no. 2025.07/0450 “Wide-bandgap gallium-based oxide materials with controlled luminescent and electrical characteristics for energy-saving optoelectronics”.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [D.S., A.L.]; methodology, [D.S., M.K., V.V.]; investigation, [D.S., M.K., V.V., I.M.]; writing – original draft preparation, [D.S., A.L.]; writing – review and editing, [D.S., M.K., V.V., A.L.]; visualization, [D.S., M.K., I.M., A.L.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Kahlenberg, V., Fischer, R. X., Parise, J. B. (2000). The stuffed framework structure of $BaGa_2O_4$. *J. Solid State Chem.*, *154*(2), 612–618. https://doi.org/10.1006/jssc.2000.8903

[2] Klym, H., Karbovnyk, I., Luchechko, A., Kostiv, Y., Pankratova, V., Popov, A. I. (2021). Evolution of free volumes in polycrystalline $BaGa_2O_4$ ceramics doped with Eu^{3+} ions. *Crystals*, *11*(12), 1515. https://doi.org/10.3390/cryst11121515

[3] Tuller, H. L., Bishop, S.R. (2011). Point defects in oxides: Tailoring materials through defect engineering. *Annu. Rev. Mater. Res.*, *41*, 369–398. https://doi.org/10.1146/annurev-matsci-062910-100442

[4] Smyth, D. M. (2000). Effects of dopants on the properties of metal oxides. *Solid State Ion.*, *129*(1-4), 5–12. https://doi.org/10.1016/S0167-2738(99)00312-4

[5] Wang Y., Su J., Lin Z., Zhang J., Chang J., Hao Y. (2022). Recent progress on the effects of impurities and defects on the properties of Ga_2O_3. *J. Mater. Chem. C*, 10(37), 13395-13436. https://doi.org/10.1039/d2tc01128j

[6] Wang, S., Wang, T., Yu, X., Li, Z., Guo, L., Chen, J., Zhao, F., Feng, W., Xu, X., Qiu, J. (2021). Tailored Luminescence Output of Bi^{3+}-Doped $BaGa_2O_4$ Phosphors with the Assistance of the Introduction of Sr^{2+} Ions as Secondary Cations. *Inorg. Chem.*, *60*(18), 14467–14474. https://doi.org/10.1021/acs.inorgchem.1c02271

[7] Luchechko, A., Vasyltsiv, V., Kushlyk, M., Syvorotka, I., Kostyk, L.; Slobodzyan, D., Zhydachevskyy, Y. (2025). Luminescence and thermally stimulated conductivity of polycrystalline β-Ga_2O_3: Mn and β-Ga_2O_3: Mn, Si. *J. Vac. Sci. Technol. A*, *43*, 042802. https://doi.org/10.1116/6.0004570

[8] Chikoidze, E., Sartel, C., Madaci, I., Mohamed, H., Vilar, C., Ballesteros, B., Belarre, F., del Corro, E., Vales-Castro, P., Sauthier, G., Li, L., Jennings, M., Sallet, V., Dumont, Y., Pérez-Tomás, A. (2020). P-Type Ultrawide-Band-Gap Spinel $ZnGa_2O_4$: New Perspectives for Energy Electronics. *Cryst. Growth Des. 20*(4), 2535–2546. https://doi.org/10.1021/acs.cgd.9b01669

[9] Linderälv, C., Lindman, A., Erhart, P. (2018). A Unifying Perspective on Oxygen Vacancies in Wide Band Gap Oxides. *J. Phys. Chem. Lett.*, *9*(1), 222–228. https://doi.org/10.1021/acs.jpclett.7b03028

[10] Shawuti, S., Can, M. M., Gülgün, M. A., Firat, T. (2014) Grain size dependent comparison of ZnO and $ZnGa_2O_4$ semiconductors by impedance spectrometry. *Electrochim. Acta*, *145*, 132–138. https://doi.org/10.1016/j.electacta.2014.08.084

[11] Luchechko, A., Vasyltsiv, V., Kushlyk, M., Kostyk, L., Slobodzyan, D. (2024). Electrical conductivity, luminescence, and deep acceptor levels in β-Ga_2O_3-In_2O_3 polycrystalline solid solution doped with Zr^{4+} or Ca^{2+} ions. *J. Vac. Sci. Technol. A*, *42*, 033202. https://doi.org/10.1116/6.0003466

[12] Swain, S. K., Weber, M. H., Jesenovec, J., Saleh, M., Lynn, K. G., McCloy, J. S. (2021). Compensation of Shallow Donors by Gallium Vacancies in Monoclinic β-Ga_2O_3. Phys. *Rev. Applied*, *15*, 054010. https://doi.org/10.1103/PhysRevApplied.15.054010
[13] Bhattacharjee, S., Ghosh, A. (2026). A critical perspective on temperature-dependent AC conductivity and its frequency exponent in the framework of non-overlapping small polaron tunnelling model. *Mater. Today Commun.*, *50*, 114584. https://doi.org/10.1016/j.mtcomm.2025.114584
[14] Jonscher, A. K. (1977) The 'universal' dielectric response. *Nature*, *267*, 673-679. https://doi.org/10.1038/267673a0
[15] Sidebottom, D. L., Murray-Krezan, C. M. (2002). Distinguishing Two Contributions to the Nearly Constant Loss in Ion-Conducting Glasses. *Phys. Rev. Lett.*, *89*, 195901. https://doi.org/10.1103/PhysRevLett.89.195901
[16] Natanzon, Y., Azulay, A., Amouyal, Y. (2020). Evaluation of Polaron Transport in Solids from First-principles. *Isr. J. Chem*, *60*(8-9), 768-786. https://doi.org/10.1002/ijch.201900101

СПЕКТРАЛЬНІ ТА ЕЛЕКТРИЧНІ ХАРАКТЕРИСТИКИ ЛЮМІНОФОРІВ $BaGa_2O_4$, ЛЕГОВАНИХ Mn ТА Mn-Zr

Дмитро Слободзян*, Маркіян Кушлик, В'ячеслав Васильців, Іванна Медвідь, Андрій Лучечко
Кафедра сенсорної та напівпровідникової електроніки, Львівський національний університет імені Івана Франка, вул. Ген. Тарнавського, 107, Львів, 79017, УКРАЇНА

АНОТАЦІЯ

Вступ. $BaGa_2O_4$ є широкозонним оксидом з перспективними люмінесцентними та електронними властивостями, що робить його потенційною матрицею для люмінофорних та оптоелектронних матеріалів. Контрольоване легування забезпечує ефективний підхід до модифікації його люмінесценції, дефектної структури та властивостей перенесення заряду.

Матеріали та методи. Нелегований $BaGa_2O_4$ (BGO), BGO: 0,1 мол.% Mn та BGO: 0,1 мол.% Mn, 0,2 мол.% Zr були синтезовані високотемпературним твердофазним методом. Спектри фотолюмінесценції та збудження фотолюмінесценції вимірювалися за кімнатної температури. Провідність постійного струму досліджувалася у вакуумі в діапазоні температур 370–490 K, тоді як провідність змінного струму вимірювалася в діапазоні частот 100 Гц–50 кГц з використанням амплітуди збуджувального електричного поля 5,0 В.

Результати. Нелегований BGO демонструє інтенсивну широкосмугову блакитну люмінесценцію з максимумом поблизу 410 нм, що пояснюється центрами випромінювання матриці. Легування Mn зміщує максимум випромінювання до 430–440 нм, значно зменшує інтенсивність та створює додаткову смугу збудження люмінесценції при 270–340 нм, що вказує на наявність ефективного каналу збудження, пов'язаного з центрами Mn. Спільне легування Zr додатково зменшує інтенсивність, але суттєво не змінює спектральний склад свічення. Згідно з аналізом CIE 1931, чистота кольору зменшується з 80% для BGO до 75% та 71% для зразків, легованих Mn і Mn-Zr відповідно. Електричні вимірювання показують, що легування Mn збільшує провідність постійного струму більш ніж на порядок, тоді як спільне легування Mn-Zr зменшує провідність, зберігаючи при цьому значення вище, ніж у нелегованого BGO. Енергії активації постійного струму залишаються близькими в діапазоні від 1,15 до

1,19 еВ. Провідність змінного струму демонструє частотну дисперсію, що описується універсальним законом Йоншера, причому температурна залежність показника *s* вказує на зміни в локалізованих механізмах переносу заряду.

Висновки. Легування Mn та Mn-Zr ефективно змінює оптичні та електричні властивості галату барію шляхом зміни локалізованих станів і дефектної структури. Здатність налаштовувати люмінесценцію та переносити заряд підкреслює потенціал легованого $BaGa_2O_4$ для застосування в люмінофорах та оптоелектроніці.

Ключові слова: $BaGa_2O_4$, домішки Mn-Zr, люмінесценція, електропровідність, локалізовані стани, люмінофори

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
08 July, 2026	27 August, 2026	31 August, 2026	28 September, 2026

Збірник наукових праць

Електроніка та інформаційні технології

Electronics and information technologies

Випуск 35

2026

Підп. до друку 30.09.2026. Формат 70x100,16. Папір друк.
Друк на різогр. Гарнітура Times New Roman. Умовн. друк. арк. .
Тираж 100 прим. Зам. № .

Львівський національний університет імені Івана Франка.
79000 Львів, вул. Університетська, 1.

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції. Серія ДК № 3059 від 13.12.2007 р.