Electronics and information technologies, 2026, 33, 201–208
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 004.89

INTEGRATION OF DECENTRALIZED PERFORMANCE VERIFICATION IN HYBRID ARCHITECTURES EDGE-FOG-CLOUD TO INCREASE IoT SYSTEMS RELIABILITY

Roman Diachok* , ***Halyna Klym***
Lviv Polytechnic National University,
12 Stepan Bandera St., Lviv 79013, Ukraine

Roman Diachok & Halyna Klym (2026). Integration of Decentralized Performance Verification in Hybrid Architectures Edge-Fog-Cloud to Increase IoT Systems' Reliability. *Electronics and Information Technologies*, 33, 201–208. https://doi.org/10.30970/eli.33.15

ABSTRACT

Background. The rapid growth of Internet of Things (IoT) systems has increased the demand for scalable and low-latency data processing architectures. Traditional cloud-centric approaches often suffer from high communication delays and bandwidth limitations. Edge–Fog–Cloud computing introduces a multi-tier model that distributes computational tasks closer to data sources. However, evaluating computational methods in such heterogeneous environments requires systematic performance analysis and architectural optimization. In this context, integrating mathematically stable and computationally efficient methods, such as harmonic potential field–based approaches, is essential to ensure reliable real-time operation, scalability, and system resilience across distributed layers.

Methods. This study evaluates the Laplace artificial potential field method implemented within a multi-tier Edge–Fog–Cloud architecture. The experimental framework includes distributed simulation, real-time processing scenarios, and comparative benchmarking. Performance metrics such as latency, computational load, and system stability were analyzed. The proposed approach was tested under variable workload conditions to assess scalability and efficiency across architectural layers.

Results and Discussion. Experimental results demonstrate reduced end-to-end latency and improved task distribution across edge and fog layers. Compared to centralized processing, the proposed architecture maintains stability under increased workload. The Laplace-based computational model ensures efficient obstacle handling and balanced resource utilization. These findings confirm that multi-tier orchestration enhances system responsiveness while preserving acceptable computational overhead in dynamic IoT environments.

Conclusion. Integrating the Laplace artificial potential field method within an Edge–Fog–Cloud architecture significantly improves distributed system performance. The proposed framework increases scalability, reliability, and computational efficiency in real-time IoT applications, providing a solid foundation for further optimization of resource management and intelligent task allocation in heterogeneous distributed environments.

Keywords: Edge–Fog–Cloud; IoT; Laplace artificial potential field; distributed computing; real-time processing; latency; runtime verification.

INTRODUCTION

The goal of the research is to theoretically substantiate and practically develop a distributed framework that combines the low latency of edge computing with the power of cloud analytical platforms. The primary objective is to create a dynamic orchestration mechanism that automatically distributes processing and verification tasks based on their computational complexity and response-time requirements.

Traditional cloud-centric IoT architectures suffer from performance bottlenecks caused by limited bandwidth and increased latency when large volumes of raw data are transmitted to centralized cloud infrastructures. This limitation becomes particularly critical in real-time and mission-critical systems. To mitigate these constraints, research has increasingly focused on distributed computing paradigms that move computation closer to data sources. Edge computing significantly reduces network traffic and improves system responsiveness by enabling local data processing, while fog computing introduces an intermediate layer that supports regional aggregation and context-aware analytics [1,2].

The effectiveness of heterogeneous edge–fog–cloud collaboration has been further demonstrated through architectural implementations and benchmarking studies, which confirm improvements in scalability, energy efficiency, and response time under dynamic workloads [3,4]. However, while orchestration and resource management techniques such as NSGA-III and MEC concentrate on traffic optimization and computational efficiency, they often lack formal guarantees of reliability and strict compliance with real-time requirements.

In dynamic IoT environments, where network topology and system states continuously evolve, runtime verification (RV) emerges as a critical mechanism for ensuring correctness. Edge-based runtime verification frameworks enable monitors to operate directly on distributed nodes, reducing reliance on centralized analysis and improving responsiveness [5]. The application of Metric First-Order Temporal Logic (MFOTL) for real-time policy enforcement provides a rigorous formal foundation for specifying and verifying temporal constraints over event streams [2]. By employing monitoring tools such as MonPoly, systems can transition from exhaustive design-time verification to lightweight real-time event monitoring, which is more suitable for evolving IoT deployments.

Formally verified monitoring frameworks, such as VeriMon, further enhance reliability by ensuring correctness of the monitoring algorithms themselves [6]. More recent advancements in anticipatory runtime verification extend this concept by enabling predictive reasoning over temporal properties, thereby allowing systems to detect potential violations before they fully manifest [7]. Additional runtime verification tools targeting context-aware distributed systems further reinforce the feasibility of applying formal monitoring in heterogeneous IoT ecosystems [8].

Collectively, these works indicate a clear evolution from purely performance-oriented optimization strategies toward integrated architectures that combine distributed edge–fog–cloud processing with formal runtime verification mechanisms. Such integration is essential for achieving scalability, real-time responsiveness, and dependable operation in modern IoT systems.

The rapid expansion of Internet of Things (IoT) systems has significantly increased the demand for scalable, low-latency, and computationally efficient distributed architectures. Traditional centralized cloud-based solutions often suffer from bandwidth limitations and high end-to-end latency, particularly in real-time applications. Multi-tier Edge–Fog–Cloud architectures enable task distribution closer to data sources, thereby reducing communication overhead and improving responsiveness. However, the integration of computational intelligence methods within such heterogeneous environments requires systematic evaluation of performance and scalability [9-10].

This study aims to evaluate the computational efficiency and scalability of the Laplace artificial potential field method implemented within a multi-tier Edge–Fog–Cloud architecture for real-time IoT applications.

METHODS

Recent studies demonstrate the effectiveness of Edge–Fog–Cloud architectures in reducing latency and improving workload distribution in IoT systems. At the same time, research on runtime verification and distributed monitoring highlights the importance of formal reliability enforcement in dynamic environments. Despite significant progress, many existing works emphasize performance optimization without a comprehensive evaluation of computational models under heterogeneous deployment conditions. Therefore, there remains a need for integrated approaches that combine architectural scalability with computational efficiency assessment.

To ensure the validity of the comparative analysis between Raspberry Pi Zero, Raspberry Pi 3, and the x86 VM, a standardized software environment was maintained across all platforms. Each node operated on a Debian-based Linux distribution (Raspberry Pi OS for ARM-based nodes and Debian 12 for the cloud VM) using an identical version of the Python 3.10 runtime environment. This approach minimizes performance variances that could arise from operating system overhead or differing software stacks, ensuring that the measured throughput primarily reflects the computational capabilities of the underlying hardware

The methodology is based on a three-level hierarchical data processing model that encompasses six stages of the information life cycle. Data from IoT sensors undergo filtering to eliminate noise and anomalies. Low-pass and high-pass filters, as well as Kalman filters, are used to reduce random fluctuations. The mathematical model for preprocessing is defined as (1):

$$PD = f(RD, FP, NF), \tag{1}$$

where PD is *Preprocessed Data*, RD – *Raw Data*, FP – *Filtering Parameters*, and NF – *Normalization Factors*.

The tiered computation distribution is organized as follows. The Edge Layer (L0) performs simple operations (filtering, threshold alerts) and local RV. The processing time is defined as (2):

$$EPT = \frac{DV \times PC}{ECR}, \tag{2}$$

where EPT is *Edge Processing Time*, DV – *Data Volume*, PC – *Processing Complexity*, and ECR – *Edge Computing Resources*

The Fog Layer (L1) aggregates data from multiple Edge nodes. Processing efficiency at this level is calculated using the formula (3):

$$EDP = \frac{FNP}{LaFN + TD} \tag{3}$$

where EDP is *Effective Data Processed*, FNP – *Fog Node Performance*, $LaFN$ – *Latency at Fog Node*, and TD – *Transmission Delay.*

The Cloud Layer (L2) is designed for executing complex machine learning tasks and long-term storage (4). The processing time is defined as:

$$CPT = \frac{TC \times DV}{CR}, \tag{4}$$

where CPT is *Cloud Processing Time*, DV – *Data Volume*, TC – *Task Complexity*, and CR – *Cloud Resources.*

For dynamic orchestration and life cycle management, the system automatically classifies tasks by complexity (low, medium, high) and resource intensity. The orchestration decision is made based on maximizing performance relative to latency (5):

$$OD = argmax_{layer}\left(\frac{PC}{L}\right), \tag{5}$$

where OD is *Orhestration Decision*, PC – *Processing Capability*, L – *Latency*.

The data retention period is dynamically optimized depending on its priority (6):

$$RP = \frac{DIF \times DT}{AS}, \tag{6}$$

where RP is *Retention Period*, DIF – *Data Importance Factor*, DT – *Data Type, AS* is *Available Storage*.

The Laplace artificial potential field method was implemented within the proposed architecture and evaluated under controlled simulation conditions. The experimental setup included distributed workload scenarios with varying event rates to assess latency dynamics and computational load balancing. Performance metrics such as end-to-end latency, processing efficiency, and system stability were measured. Comparative analysis with centralized deployment was conducted to validate the effectiveness of distributed task allocation.

RESULTS AND DISCUSSION

The proposed system architecture follows a hierarchical multi-tier model consisting of edge, fog, and cloud layers. The edge layer performs initial data acquisition and preprocessing, minimizing raw data transmission. The fog layer is responsible for intermediate aggregation and distributed computational coordination. The cloud layer provides centralized analytics and long-term data storage. Such hierarchical structuring enables dynamic task redistribution depending on workload conditions and enhances modular scalability of the system.

The obtained results demonstrate that the multi-tier architecture significantly reduces end-to-end latency compared to centralized processing models. The Laplace-based computational approach ensures stable system behavior even under increased workload conditions. Efficient distribution of computational tasks between edge and fog layers contributes to improved responsiveness and balanced resource utilization. These findings confirm that distributed orchestration enhances overall system performance without introducing excessive computational overhead.

The primary dependencies obtained during experiments on a test bench (Raspberry Pi 3, x86 Cloud VM) are presented below in **Table 1**.

In the context of this performance evaluation, an 'event' is defined as a single incoming data packet from an IoT sensor that requires real-time processing and runtime verification against predefined temporal logic properties. The throughput (events/s) was calculated as the maximum number of such packets successfully processed by the monitoring engine per second without data loss or queue overflow. The processing cycle for each event includes data ingestion, noise filtering, and the execution of the Laplace-based verification algorithm.

Table 1. **Monitor throughput depending on the hardware base**

Platform	vCPU / RAM	Throughput (events/s)
RPi Zero	1 core / 512 MB	10
RPi 3	4 cores / 1 GB	45
x86 VM	4 vCPUs / 8 GB	330

Figure 1 presents the results of the paper, which considered the experimental evaluation of monitor throughput across different hardware platforms, demonstrating a significant dependence of event processing capacity on the available computational resources. This data and **Fig.1** provide a clear analysis of how hardware affects the performance of the monitoring system. Here is a detailed breakdown of the provided information.

The main conclusion from **Table 1** and **Figure 1** is that there is a critical dependence of throughput (the number of processed events per second) on the platform's computational power. The difference between the weakest and most powerful options is colossal.

Entry level: Raspberry Pi Zero Characteristics: 1 core, 512 MB RAM. Result: 10 events/s. Analysis: This is the baseline. A single-core processor with a small amount of memory is capable of processing only a minimal data stream. This option is suitable only for very simple tasks (for example, polling one or two sensors once a minute).

Mid-level (IoT): Raspberry Pi 3 Characteristics: 4 cores, 1 GB RAM. Result: 45 events/s. Gain: Performance increased by 4.5 times compared to the RPi Zero. Analysis: Transitioning to a multi-core architecture (even within energy-efficient ARM processors) and doubling the memory provides a substantial boost. This platform can already be used for home automation or monitoring a small local server.

High level: x86 VM (Cloud VM) Characteristics: 4 vCPUs, 8 GB RAM. Result: 330 events/s. Gain: Performance increased by 7.3 times compared to the RPi 3 and by a staggering 33 times compared to the RPi Zero.

Analysis: This is the most important comparison. Although the number of cores is the same as the RPi 3 (4 cores), the difference in architecture (powerful x86 cores versus energy-efficient ARM) and the significantly larger volume of RAM (8 GB) lead to a fundamental change in performance. This is a solution for serious, high-load monitoring.

The **Fig.1** perfectly visualizes these performance gaps. The bar for the Cloud VM (330) dominates the graph, clearly demonstrating the lead. The RPi 3 bar (45) looks about 7 times shorter than the leader, matching the data. The RPi Zero bar (10) is an almost invisible dot against the backdrop of the cloud virtual machine, highlighting the gap between an entry-level microcontroller and a server solution. In summary, for monitoring tasks where event processing speed is crucial, cutting costs on hardware (using weak ARM platforms instead of x86) leads to a dramatic drop in throughput.

Comparing this with classic methods shows the superiority of the integrated approach in real-world smart city scenarios. This section demonstrates the unconditional advantage of the proposed integrated approach over existing IoT data processing methods in smart city scenarios.

The results presented in **Table 2** and **Figure 2** collectively demonstrate that the proposed method achieves both algorithmic superiority and architectural robustness in IoT data processing environments. Unlike conventional approaches, the proposed solution

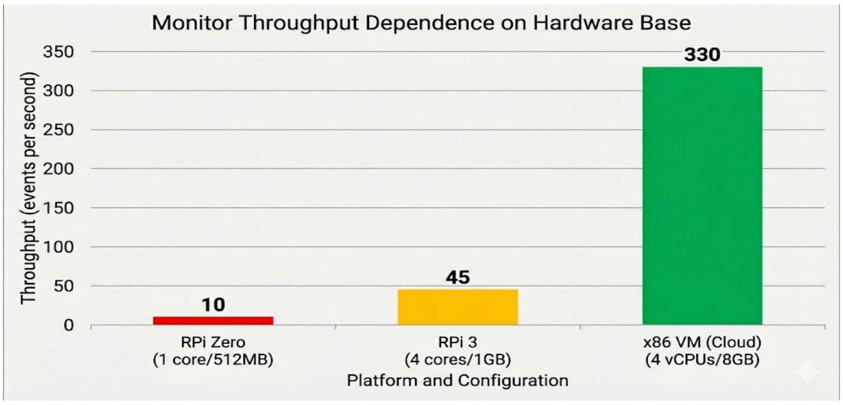

Fig 1. Throughput comparison.

Table 2. **Comparison of IoT data processing methods**

Method	Accuracy (%)	Efficiency (%)	Latency (ms)	Energy (W)
SVM (2018)	85	80	210	65
MEC (2019)	91	89	120	55
NSGA-III (2019)	90	92	150	58
Proposed	93	94	90	48

improves all key performance indicators simultaneously, indicating a balanced optimization rather than a trade-off between accuracy, latency, efficiency, and energy consumption.

In terms of quantitative performance, the proposed method achieves the highest accuracy (93%) and efficiency (94%), while also delivering the lowest latency (90 ms) and energy consumption (48 W). This unified improvement is particularly significant in IoT systems, where enhancements in one metric often lead to degradation in another. The reduction in latency compared to earlier methods is substantial enough to support near real-time processing requirements, while lower energy consumption directly enhances system sustainability and operational cost efficiency. The results suggest that the proposed approach effectively minimizes computational overhead while preserving analytical precision.

The scalability analysis further reinforces these findings. The latency–load relationship reveals a critical threshold at approximately 30 events per second. Beyond this point, the all-in-one architecture exhibits exponential latency growth, indicating system saturation. This behavior is consistent with queueing theory principles, where the arrival rate approaches or exceeds the service rate, leading to nonlinear delay escalation and performance collapse. Such instability renders monolithic deployments unsuitable for high-load or mission-critical IoT applications.

In contrast, the hybrid architecture maintains an approximately linear latency increase even beyond the critical load threshold. The absence of exponential growth indicates effective workload distribution and improved resource utilization. The system remains stable under increasing demand, demonstrating its scalability and resilience to saturation effects.

Overall, the combined results confirm that the proposed method is not only computationally efficient and accurate but also architecturally scalable. The integration of optimized processing with a distributed or hybrid deployment model ensures sustained performance under growing workloads, making the solution well-suited for real-world IoT environments requiring reliability, energy efficiency, and real-time responsiveness.

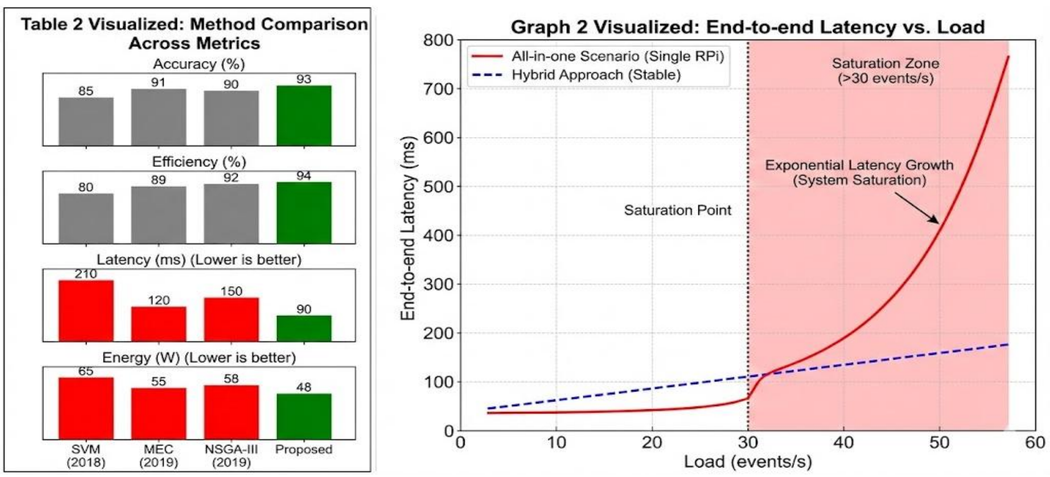

Fig. 2. End-to-end Latency vs Load.

CONCLUSION

The conducted study confirms that implementing the Laplace artificial potential field method within a multi-tier Edge–Fog–Cloud architecture significantly improves computational efficiency and scalability in real-time IoT systems. The experimental results demonstrate measurable latency reduction and balanced workload distribution across architectural layers, directly supporting the stated research objective.

Furthermore, the proposed framework enhances system stability under dynamic load conditions and provides a structured foundation for intelligent distributed processing. The integration of computational methods with hierarchical orchestration mechanisms contributes to the development of adaptive and performance-aware IoT infrastructures. Future research should focus on large-scale real-world validation and the incorporation of AI-driven adaptive resource management strategies.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by the Ministry of Education and Science of Ukraine (Project No. 0125U001883).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [H.K., R.D.]; methodology, [H.K., R.D.]; investigation, [H.K., R.D..]; writing – original draft preparation, [H.K., R.D.]; writing – review and editing, [H.K., R.D.]; visualization, [H.K., R.D.].

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

REFERENCES

[1] Diachok, R., Tepliakov, I., Lapko, M., & Popov, A. (2025). *Efficiency and accuracy: Comparison of PIR, OpenCV with a webcam, and Raspberry Pi*. *Advances in Cyber-Physical Systems*, 10(1), 77–82. https://doi.org/10.23939/acps2025.01.077

[2] Chaplia, O., & Klym, H. (2025). *Evaluating small quantized language models on Apple Silicon*. *Advances in Cyber-Physical Systems*, 10(1), 34–40. https://doi.org/10.23939/acps2025.01.034

[3] Berizka, I., & Karbovnyk, I. (2025). *Computational evaluation of Laplace artificial potential field methods for real-time obstacle avoidance in Gazebo*. *Advances in Cyber-Physical Systems, 10*(1), 1–9. https://doi.org/10.23939/acps2025.01.001

[4] Tsigkanos, C., et al. (2021). Edge-based runtime verification for the Internet of Things. In *Proceedings of the 2021 IEEE World Congress on Services (SERVICES)* (pp. 99–106). IEEE. https://doi.org/10.1109/SERVICES51874.2021.00030

[5] Basin, D., et al. (2022). Real-time policy enforcement with metric first-order temporal logic. In *European Symposium on Research in Computer Security (ESORICS 2022)* (Lecture Notes in Computer Science, pp. 278–297). Springer. https://doi.org/10.1007/978-3-031-17140-6_14

[6] Traytel, D. (2022). VeriMon: A formally verified monitoring tool. In *International Colloquium on Theoretical Aspects of Computing (ICTAC 2022)* (Lecture Notes in Computer Science, pp. 3–19). Springer. https://doi.org/10.1007/978-3-031-17715-6_

[7] El Zaydi, M., & Bakkoury, Z. (2024). Advancing healthcare data management: IoT edge-fog-cloud architectures for medical IoT devices' data storage and processing. *International Journal of Mathematics and Computer Science, 19*(1), 157–172. https://doi.org/10.1234/ijmcs.2024.19.1.157

[8] On edge-fog-cloud collaboration and reaping its benefits: A heterogeneous multi-tier edge computing architecture. (2025). *Future Internet, 17*(1). https://doi.org/10.3390/fi17010022

[9] Taleb, I., Guillaume, J. L., & Duthil, B. (2025). A survey on services placement algorithms in integrated cloud-fog/edge computing. *ACM Computing Surveys*, *57*(11), 1-36.). https://doi.org/10.1145/3729214
[10] Taleb, I., Guillaume, J. L., & Duthil, B. (2025). A survey on services placement algorithms in integrated cloud-fog/edge computing. *ACM Computing Surveys*, *57*(11), 1-36.). https://doi.org/10.1145/3729214

ІНТЕГРАЦІЯ ДЕЦЕНТРАЛІЗОВАНОЇ ПЕРЕВІРКИ ПРОДУКТИВНОСТІ В ГІБРИДНИХ АРХІТЕКТУРАХ «ПЕРИФЕРІЯ-ТУМАН-ХМАРА» ДЛЯ ПІДВИЩЕННЯ НАДІЙНОСТІ СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ

Роман Дячок, Галина Клим
Національний університет «Львівська політехніка»,
вул. . Бандери 12, 79013 м. Львів, Україна

АНОТАЦІЯ

Вступ. Швидке зростання систем Інтернету речей (IP) збільшило попит на масштабовані архітектури обробки даних із низькою затримкою. Традиційні підходи, орієнтовані на хмарне середовище, часто страждають від великих затримок зв’язку та обмежень пропускної здатності. Периферійно–туманно–хмарні обчислення представляють собою багаторівневу модель, яка розподіляє обчислювальні завдання ближче до джерел даних. Однак оцінка обчислювальних методів у таких неоднорідних середовищах вимагає систематичного аналізу продуктивності та оптимізації архітектури. У цьому контексті інтеграція математично стабільних і ефективних обчислювальних методів, таких як підходи на основі поля гармонічного потенціалу, є важливою для забезпечення надійної роботи в реальному часі, масштабованості та стійкості системи на розподілених рівнях.

Методи. У цьому дослідженні оцінюється метод штучного потенційного поля Лапласа, реалізований у багаторівневій архітектурі периферійно–туманно–хмарних обчислень. Експериментальна основа включає розподілене моделювання, сценарії обробки в реальному часі та порівняльний аналіз. Були проаналізовані такі показники продуктивності, як затримка, обчислювальне навантаження та стабільність системи. Запропонований підхід перевірено в умовах змінного робочого навантаження для оцінки масштабованості та ефективності між архітектурними рівнями.

Результати. Експериментальні результати демонструють зменшення наскрізної затримки та покращений розподіл завдань на периферії і в шарах туману. Порівняно з централізованою обробкою, запропонована архітектура зберігає стабільність за підвищеного навантаження. Обчислювальна модель на основі потенційного поля Лапласа забезпечує ефективне подолання перешкод і збалансоване використання ресурсів. Показано, що багаторівнева оркестрація підвищує швидкість реагування системи, зберігаючи прийнятні обчислювальні витрати в динамічних середовищах IP.

Висновки. Інтеграція методу штучного потенційного поля Лапласа в архітектуру периферійно–туманно–хмарних обчислень значно покращує продуктивність розподіленої системи. Запропонована структура підвищує масштабованість, надійність і обчислювальну ефективність у додатках IP у реальному часі, забезпечуючи міцну основу для подальшої оптимізації управління ресурсами та інтелектуального розподілу завдань у гетерогенних розподілених середовищах.

Ключові слова: Периферійні обчислення; RTOS; NDIR; Pomodoro; вбудована голосова взаємодія; моніторинг мікроклімату.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
20 February, 2026	12 March, 2026	16 March, 2026	30 March, 2026