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

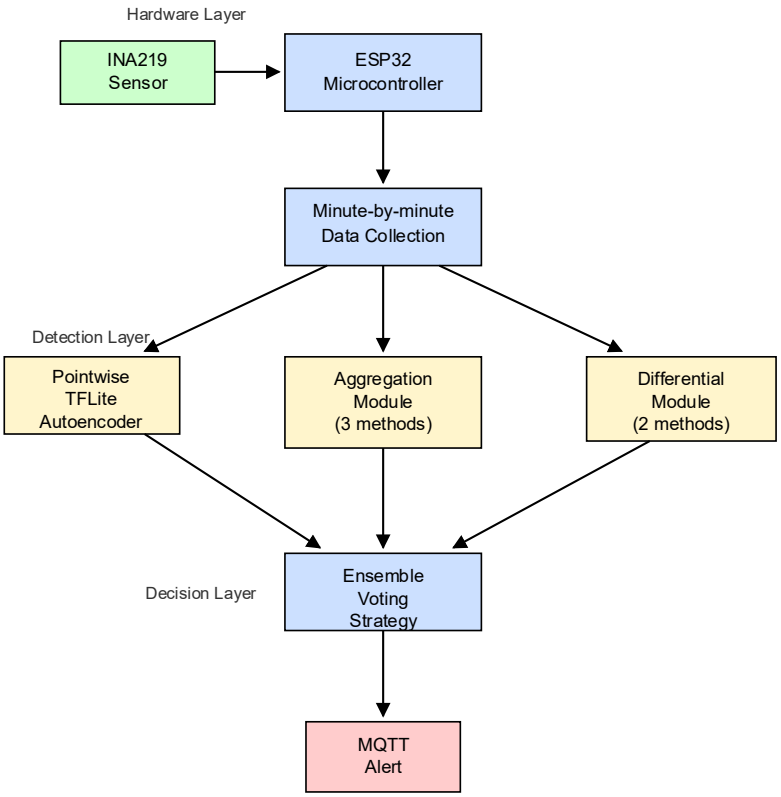


Fig. 1. Structural flowchart of the multi-tiered edge AI anomaly detection architecture

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