Detecting anomalous activity in smart grids to analyze node energy profiles

Inna Rozlomii[1,2,*,†], Andrii Yarmilko[2,†] and Serhii Naumenko[2,†]

[1] *Cherkasy State Technological University, 460, Shevchenko Blvd., Cherkasy, 18006, Ukraine*
[2] *Bohdan Khmelnytsky National University of Cherkasy, 81, Shevchenko Blvd., Cherkasy, 18031, Ukraine*

Abstract

This paper presents a lightweight anomaly detection method for smart networks based on energy profiling of low-power nodes. The approach targets resource-constrained devices such as sensors and microcontrollers, where traditional intrusion detection systems (IDS) are impractical due to high computational or energy demands. By continuously monitoring current, voltage, and power consumption, a baseline energy profile is constructed and used to identify deviations indicative of malicious activity. The method was implemented on ESP32-S3 microcontrollers equipped with INA219 sensors and evaluated under various operational scenarios including normal behavior and simulated attacks such as excessive wireless transmission, forced wake-ups, command injection, and battery-drain patterns. Experimental results demonstrate high detection accuracy (93.8%) and sensitivity (91.2%) with minimal overhead on CPU and memory usage. The circular buffer structure and real-time processing allow integration directly into the node's firmware, enabling autonomous operation without reliance on external infrastructure. Compared to signature- or traffic-based IDS approaches, energy profiling offers a passive, hardware-supported alternative that is particularly well-suited for decentralized, energy-aware smart environments. This work also outlines limitations such as sensitivity to short-term attacks and sensor calibration drift, and suggests future improvements including adaptive profiling, multi-parameter fusion, and distributed detection within clusters

Keywords

smart networks, anomaly detection, energy profiling, low-power IoT, embedded security, intrusion detection

1. Introduction

In recent years, there has been a rapid increase in the number of attacks on energy-efficient devices operating as part of smart grids [1]. Such devices include sensors, executive modules and microcontrollers. They are characterized by limited computing power, limited memory and, as a rule, autonomous power supply [2-4]. Their mass implementation in critical infrastructure, industrial systems, logistics platforms, "smart" homes and medical applications makes these devices an attractive target for attacks [5]. The vulnerability of such components is enhanced by the fact that classic information security tools are excessively resource-intensive for their use on such platforms [6]. This limits the possibilities of using standard intrusion detection mechanisms (IDS), which are usually based on deep traffic analysis, signatures or machine learning. Against the background of these limitations, approaches that use side characteristics of device behavior become particularly relevant. One of such characteristics is the power consumption profile. Abnormal deviations in the energy consumption of nodes can be an indicator of suspicious activity or interference by an external object. For example, malicious software or network attacks can cause atypical fluctuations in the device's operating modes, wake-up frequency, or data transfer volumes. Thus, energy profile analysis opens up the possibility of creating lightweight monitoring mechanisms that can be implemented without a significant load on the computing resources of smart grid components.

CIAW-2025: Computational Intelligence Application Workshop, September 26-27, 2025, Lviv, Ukraine

[1*] Corresponding author.

[†] These authors contributed equally.

inna-roz@ukr.net (I. Rozlomii); a-ja@ukr.net (A. Yarmilko); naumenko.serhii1122@vu.cdu.edu.ua (S. Naumenko)
0000-0001-5065-9004 (I. Rozlomii); 0000-0003-2062-2694 (A. Yarmilko); 0000-0002-6337-1605 (S. Naumenko)

Figure 1 shows the dynamics of the growth of attacks on energy-saving devices in the period from 2018 to 2025 according to independent cybersecurity think tanks [7]. Visually, a stable trend towards an increase in the number of attacks is observed, which further confirms the relevance of the chosen research direction.

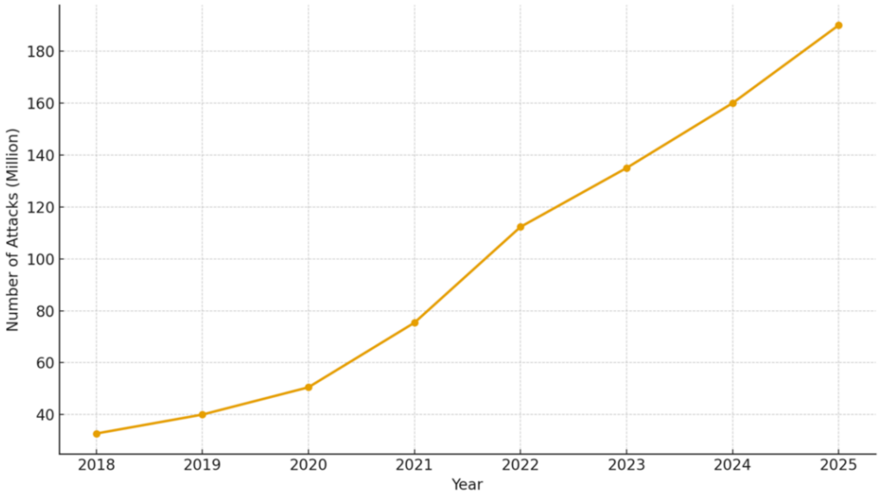


Figure 1: Graph of the growth in the number of attacks on low-power devices in 2018–2025.

Therefore, the relevance of the task of detecting anomalies based on energy characteristics is due to the need for non-load-bearing, adaptive protection methods that do not require complex calculations or centralized traffic analysis. This approach allows for a basic check of the device status even in autonomous or decentralized operation scenarios where traditional monitoring tools are not available. In addition, node energy profiles can act as a universal indicator that is common to different microcontroller architectures and protocol types, which increases the flexibility of the proposed solution.

Of scientific interest is the fact that energy consumption analysis allows for the detection of not only attacks related to data transmission or reception, but also software anomalies that cause incorrect node functioning (for example, infinite loops, unwanted awakenings from sleep mode, blocking of the main processing cycle). Thus, the approach has the potential to detect a wide range of threats to the functioning of smart networks, including zero-day attacks.

In view of the above, the purpose of the article is to study a method for detecting anomalous activity in smart networks by analyzing the energy profiles of nodes with subsequent evaluation of its effectiveness on an experimental stand.

2. Theoretical background and related works

The issue of protecting energy-saving devices in smart grids is traditionally considered through the prism of intrusion detection systems, which are conventionally divided into three main categories: signature-based, anomaly-based, and hybrid. Signature methods use known attack patterns to compare with current traffic or device behavior. These systems are characterized by high accuracy in detecting already known threats, but are completely ineffective against unknown or modified attacks [8]. Behavioral methods, on the contrary, are based on detecting deviations from the standard behavior of the system, which makes it possible to respond to zero-day attacks, but is accompanied by a high level of false positives [9]. Hybrid approaches try to combine the advantages of both types, but their implementation is much more resource-intensive, which makes their widespread use in low-power systems impossible [10]. At the same time, a separate area of research focuses on monitoring energy consumption as a key side characteristic that reflects the overall state of the device [11]. Studies have shown that analyzing energy profiles in WSNs and IoT networks can detect both node failures and potential attacks, including DoS, battery-draining, and replay attacks [12]. In [13], [14], an approach to intrusion detection is proposed by calculating the average and peak energy consumption in certain time windows and comparing these values with an acceptable baseline formed during the training process. However, most existing implementations are either based on simulations or require additional hardware, such as external current sensors, which is not always acceptable in real conditions. The disadvantages of classical IDS systems in the context of IoT architectures and smart networks are well documented: they often have excessive energy consumption, require a stable communication channel with the processing server, and consume a large amount of memory [15]. Most typical IoT platforms (STM32, ESP32, nRF52) have only a few tens of kilobytes of RAM, which excludes the possibility of placing voluminous analytical models or logs on the device side. In addition, wireless communication in such networks is usually implemented based on energy-saving protocols that do not guarantee data integrity in real time, which further complicates the construction of classic IDS.

The scientific literature also demonstrates a growing interest in using side characteristics of the functioning of autonomous devices, such as response time, thermal fingerprints, electromagnetic signatures or acoustic profiles. In particular, the idea of using such parameters to detect side channel attacks or unauthorized changes to the device firmware is discussed in [16], [17]. However, here too there is a problem of energy consumption and implementation complexity. Analyzing the existing approaches, we can conclude that most of them are not suitable for use on devices with limited resources without significant adaptation.

In this regard, the combination of the energy profiling method with a lightweight anomaly detection mechanism that does not require complex data processing, preserves the accuracy of threat detection and ensures implementation on resource-limited hardware platforms looks promising and is new in the scientific sense. The proposed approach allows for local activity analysis without the need to send a large amount of telemetry to cloud services, which significantly reduces energy consumption and threat detection delays.

The scientific novelty of the proposed approach lies in the development of a lightweight, node-level anomaly detection method that utilizes real-time energy profiling to identify abnormal behavior in smart grid environments. Unlike existing solutions, this method does not rely on external infrastructure or complex analytics, enabling its integration directly into resource-constrained devices. The solution introduces a structured three-layer architecture that operates autonomously on microcontroller platforms with limited memory and processing power. Additionally, the implementation of a ring-buffer–based energy history and the use of side-channel indicators such as power deviation contribute to its novelty, offering a practical, hardware-efficient alternative to traditional IDS systems.

3. Anomaly detection method

The proposed method for detecting anomalies in smart grids is based on the principle of analyzing the energy profile of a node, taking into account changes in its operating modes. The main idea is to form a normal energy profile for each node in the system training phase and then compare the actual energy consumption indicators with the permissible limits during operation. This approach allows you to record atypical activity that may be caused by external interference or an internal failure.

The normal energy consumption profile is formed in the process of continuous monitoring of the node under the conditions of its normal operation. For each of the phases of the device's activity – sleep mode, data transmission, signal processing, waiting – a set of statistical indicators is calculated. The main metrics are:

- instantaneous energy consumption (power at a specific moment in time);
- average value for a certain period (using a sliding window);
- accumulated value of energy consumption (power integral over a time interval).

For each type of activity, an acceptable range of fluctuations is determined, which takes into account the baseline and the noise threshold, which depends on hardware fluctuations, load and measurement error. Anomaly detection occurs when the observed value goes beyond the permissible interval with a certain frequency or continues beyond the set time threshold.

To ensure local analysis of energy consumption, each node must contain an architecturally simple but functional circuit that includes:

- a current or voltage measurement module (for example, a built-in ADC or an external sensor such as INA219);
- an energy profile buffer that stores a short-term history of energy indicators;
- a lightweight analysis module that implements a check for compliance with the permissible profile.

The generalized structure of the proposed solution is presented in Figure 2. The architecture assumes the presence of an energy measurement module, a profile storage buffer and a lightweight analysis module that operate directly on the device. This approach allows for local inspection for deviations from the normal energy profile without transmitting large amounts of data to the network. This creates conditions for balancing the accuracy of anomaly detection and resource efficiency, which is especially important for energy-constrained devices in smart grids.

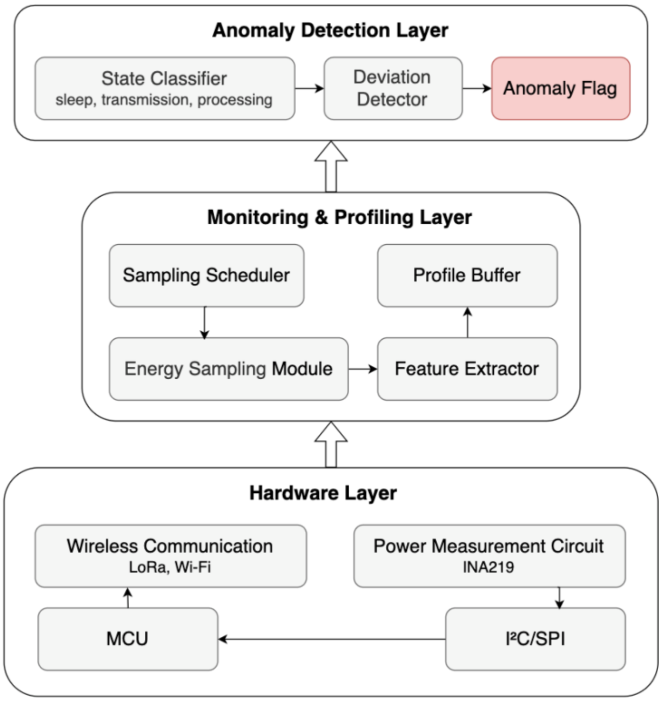


Figure 2: Smart grid node architecture with energy profile analysis module for anomaly detection.

The architecture described in Figure 2 demonstrates the internal organization of a smart grid node with a built-in anomaly detection module based on energy profile analysis. It is structured in the form of three functional layers: Hardware Layer, Monitoring & Profiling Layer, and Anomaly Detection Layer. These layers allow for a logical separation of computational and analytical functions according to their resource intensity and role in the processing process.

The Hardware Layer contains the basic physical components of the node: a microcontroller (MCU), wireless communication modules (e.g., LoRa, Wi-Fi), a power measurement circuit (Power Measurement Circuit) with an INA219 sensor, and a power supply. The sensor measures the current or voltage at the node input, after which the data is sent to the controller via the I²C or SPI digital interface for further processing.

At the Monitoring & Profiling Layer level, the logic for converting raw measurements into useful statistical characteristics is implemented. The Energy Sampling Module operates under the control of the Sampling Scheduler, after which the calculated characteristics – instantaneous power, average value and accumulated consumption – are transmitted to the Feature Extractor module. The results are stored in the Profile Buffer, where they are accumulated within a sliding window for further analysis.

Finally, the Anomaly Detection Layer is responsible for analyzing the energy profile and detecting deviations. The State Classifier module determines the current phase of node activity (e.g., sleep, transmission, processing), after which the Deviation Detector compares the obtained values with the corresponding template from the Normal Profile Model. If a stable deviation from the permissible limits is detected, an Anomaly Flag signal is generated, which can be used for local reaction or transmission to the central node. This architecture provides efficient, autonomous, and lightweight anomaly detection without the need to engage external analytical resources.

The implementation of the method involves certain assumptions and limitations. In particular, the hardware platform must support the measurement of instantaneous or average current/voltage with sufficient frequency and accuracy. In the case of using external sensors (for example, INA219), additional errors are possible, which must be taken into account when setting the thresholds. It is also assumed that the node operates mainly in repetitive activity cycles, which allows for stable formation of reference ranges.

The method does not involve the transmission of raw energy data to the network, which ensures low load on the communication channel and allows the solution to be integrated even into weakly connected or autonomous systems. The next section will describe the process of implementing the bench, processing algorithms, and testing parameters to evaluate the effectiveness of the method.

4. Implementation of the experimental stand

4.1. Hardware and software implementation of energy monitoring

To experimentally verify the proposed approach, a bench was created that simulates the operation of a typical smart grid node with limited resources. The main goal of the implementation is to test the functionality of local energy consumption monitoring, energy profile formation, and abnormal activity detection without using external servers or cloud analytics. The bench is built in such a way as to provide flexible configuration of device operating modes, simulation of typical load scenarios, and the possibility of fault injection.

The hardware part is implemented based on the ESP32-S3 microcontroller, which has built-in support for Wi-Fi and Bluetooth Low Energy (BLE) wireless communication (supporting IEEE 802.11 b/g/n Wi-Fi standards), sufficient memory (512 KB SRAM, up to 8 MB PSRAM), and built-in low-power modes (deep sleep, light sleep) [18]. The platform also supports a built-in analog-to-digital converter (ADC), which can be used for direct voltage measurement, which simplifies the basic assessment of energy consumption. ESP32-S3 is affordable, supported by open SDKs (ESP-IDF, Arduino Core) and allows flexible implementation of both monitoring and data processing algorithms without the need for additional processors [19].

To improve the accuracy of the measurement, the use of external sensors INA219 and HLW8032, which provide hardware control of the consumed power or energy, as well as the built-in ADC ESP32, was considered. A comparison of the characteristics of these components is given in Table 1.

The INA219 was chosen as the main sensor for the experiment because it provides high accuracy of current measurement through a shunt resistor, is easy to integrate via I²C, and has ready-made libraries with support for data scaling. The HLW8032 allows you to receive already calculated values of active energy via UART, but requires more complex configuration and calibration, which complicates its use in a flexible experimental environment [20].

Table 1
Characteristics of sensors for hardware control of the consumed power or energy

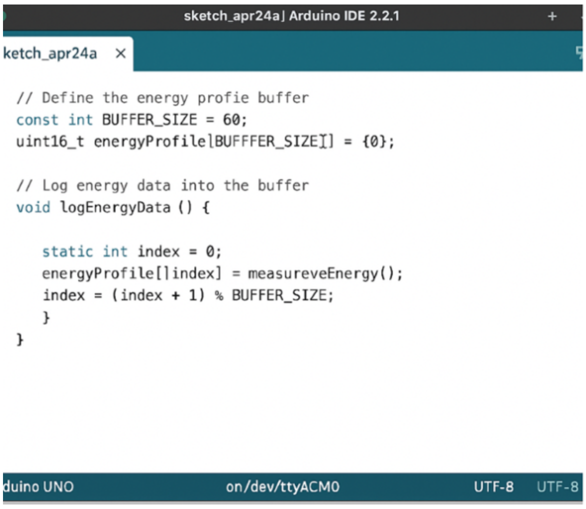
Parameter	INA219	HLW8032	ESP32 built-in ADC
Type	Current and voltage sensor	Voltage/Current/Power Sensor	ADC
Interface	I²C	UART	GPIO (built-in)
Resolution	12 bits	24 bits	12 bits
Current range	±3.2 A (depending on shunt)	up to 10 A	depends on the scheme
Supply voltage	3.0–5.5 V	3.3–5 V	3.0–3.6 V
Measured parameters	Voltage, current, power	Voltage, current, energy	High-voltage
Features	High precision, open driver	Ready-made active energy calculations	Easy integration

The sensor is connected to the ESP32 via the I²C interface (SDA/SCL lines), operates at 3.3 V, and the INA219 library with support for current sampling, voltage, and instantaneous power calculation is used on the controller side [21]. A reference load and a laboratory multimeter were used for calibration, which allowed adjusting the current scaling factor in the sensor formula.

A typical device operation cycle involves alternating the following phases: periodic awakening from deep sleep mode, reading data from the sensor, transmitting a packet via Wi-Fi or BLE, saving data to a buffer and returning to sleep mode. This mode allows you to accurately track the dynamics of energy consumption when the load changes and allows you to distinguish anomalies associated with unplanned awakenings, retransmission or background processes.

The software implementation of the test bench components is carried out in C++ using the Arduino IDE environment. The INA219 library is used to read instantaneous current, voltage and power calculations. The algorithm implements sampling with a frequency of 1 Hz, as well as calculating the average and accumulated value for a 10-second sliding window. The profile parameters are stored in an array of 60 records, which corresponds to one minute of active analysis. Figure 3 shows a fragment of the Arduino IDE code that implements the declaration of the energy profile buffer and the logic for updating it in real time.

The energy profile buffer is implemented as a ring array in the device's RAM. Each record contains a timestamp, the value of the instantaneous current, voltage, power, and the flag of the current activity phase. If the permissible range of values is exceeded, the profile is marked as suspicious and transferred to the log. The results can be saved both in CSV format (when transferred via UART to a computer) and in the device's internal memory for offline logging. The ring structure of the buffer allows you to efficiently store a limited number of recent records without exceeding the available memory limit. Figure 4 shows an example of the organization of such a buffer with a demonstration of the fields of each record and the principle of cyclic updating.

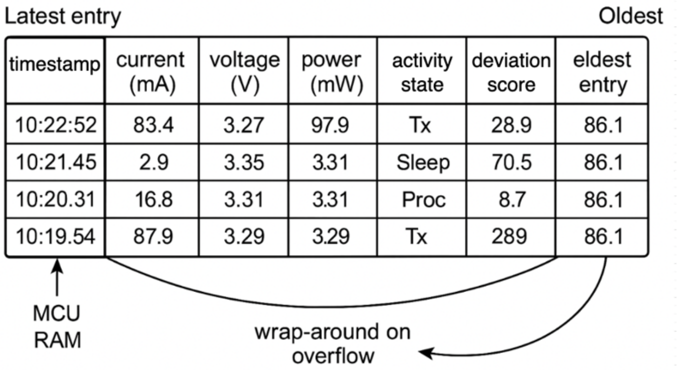


```
// Define the energy profie buffer
const int BUFFER_SIZE = 60;
uint16_t energyProfile[BUFFFER_SIZE[]] = {0};

// Log energy data into the buffer
void logEnergyData () {

   static int index = 0;
   energyProfile[]index] = measureveEnergy();
   index = (index + 1) % BUFFER_SIZE;
   }
}
```

Figure 3: Structure of the energy profile buffer and example code for data logging in Arduino IDE.



timestamp	current (mA)	voltage (V)	power (mW)	activity state	deviation score	eldest entry
10:22:52	83.4	3.27	97.9	Tx	28.9	86.1
10:21.45	2.9	3.35	3.31	Sleep	70.5	86.1
10:20.31	16.8	3.31	3.31	Proc	8.7	86.1
10:19.54	87.9	3.29	3.29	Tx	289	86.1

Figure 4: Circular buffer structure in RAM for energy profiling.

As can be seen from Figure 4, the ring buffer in the RAM contains structured records with key parameters of the node's power consumption at specific points in time. Each record records the values of current, voltage, calculated instantaneous power, device activity state (for example, transmission, processing, or sleep), as well as a deviation index from the baseline profile. This index allows you to quickly assess how well the node's current behavior matches the expected mode. An elevated value, such as 289, indicates a potentially dangerous deviation – for example, during a battery-drain attack – while a value of 8.7 is typical of a normal power mode. This structure allows you to effectively localize anomalous periods and respond quickly without complex calculations.

4.2. Anomaly modeling and log processing

To evaluate the effectiveness of the developed approach, a series of experiments was implemented, covering both normal node operation scenarios and artificially simulated abnormal situations. The purpose of these experiments is to investigate the response of the anomaly detection system to various types of violations typical of the real smart grid environment.

In normal mode, the node operates in an energy-saving scenario cycle, where the main activity occurs periodically. In particular, it is provided to wake up every 10 seconds, read a conditional sensor, transmit a packet via Wi-Fi and return to deep sleep mode. Under such conditions, the power consumption parameters are stable, and the profile is characterized by typical instantaneous power levels, which reflect a short phase of activity and a long period of sleep with minimal current.

To assess the sensitivity of the system, several attack scenarios were implemented. The first scenario simulated excessive activity on the air, when the node forcibly initiates data transmission at an increased frequency, which leads to increased power consumption in the Tx phase. The second scenario simulated frequent wake-ups due to external interrupts or invisible timers, as a result of which the device consumes more power due to shortened sleep phases. The third scenario involved injection of third-party commands via UART or Wi-Fi, which caused unexpected operations to be performed. The last type of violation is a battery-drain attack, when an attacker intentionally causes maximum load on the node, forcing it to be in an active state for a long time or to perform continuous cycles. For each of these cases, the system generates a log file containing a series of records with power consumption parameters and corresponding state flags. Depending on the configuration, the log can be stored locally in the controller's internal memory (SPI Flash) or transferred to a computer via UART for saving in CSV or JSON format. The records contain time stamps, measured values of current, voltage, power, as well as the result of the classification of the current phase and a mark about the detected anomaly.

Further analysis of the logs is carried out in two stages. The first is local processing at the MCU level using a moving average and threshold checking, which allows detecting basic deviations. The second is detailed analysis on the PC side. The data is exported to the Python environment using the pandas and matplotlib libraries, where their visualization is provided in the form of energy consumption graphs on a timeline with marking of periods in which anomalies were recorded. A preliminary classification of activity periods was also used to filter out normal modes before analysis.

The results of such experiments allowed not only to identify the most characteristic patterns of behavior of the device under study during attacks, but also to verify the correct functioning of the detection mechanism based on the energy profile. In the next section, these results will be summarized and presented in numerical form to assess the accuracy, performance, and resource intensity of the proposed solution.

5. Performance evaluation

To quantitatively assess the effectiveness of the proposed method, a series of experiments were conducted in a laboratory environment. The experimental network consisted of 5 nodes based on ESP32-S3 microcontrollers, each of which was equipped with an INA219 sensor for measuring the current consumption. The duration of each observation session was 6 hours, with alternating switching between normal scenarios and phases of simulated attacks. The nodes operated under conditions of a stable wireless Wi-Fi connection in an isolated test area with a limited background radioactivity, a temperature of 22–24°C, a constant supply voltage level (3.3V) and a standard test load in the form of sensor measurement emulation.

During the normal mode, a stable alternation of sleep and activity phases was recorded with the expected power consumption values: ≈ 270–300 mW during data transmission, ≈ 80–90 mW in the processing phase and less than 10 mW in the deep sleep mode. At the same time, during the attacks, characteristic anomalies were observed: a sudden increase in power to 350–400 mW, a violation of the cyclicity of the profile, a reduction in the duration of the sleep phase, an increase in the frequency of transmissions or the appearance of long-term phases with unstable power consumption levels. Figure 5 and Figure 6 show power consumption graphs in the normal mode and during the battery-drain attack, respectively.

The graph in Figure 5 demonstrates the regular cyclicity of the node’s power consumption during normal operation. Every 10 seconds, a power peak of ≈ 280–290 mW is observed, which corresponds to the data transfer phase. Within 2–3 seconds after the peak, the power decreases to ≈ 80–90 mW, which corresponds to the processing phase. After this, a deep drop to 5–10 mW occurs – this is the deep sleep mode, which lasts until the beginning of the next cycle. Such a characteristic profile with clear phases of activity and sleep indicates normal, predictable behavior of the node without external interference.

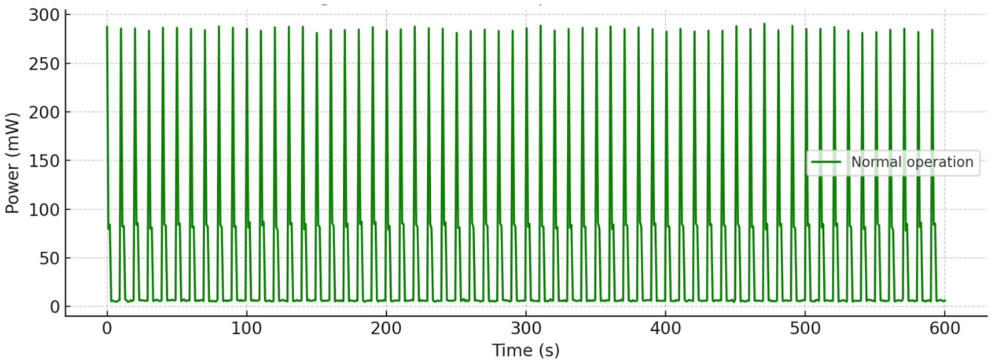


Figure 5: Normal operation energy consumption profile.

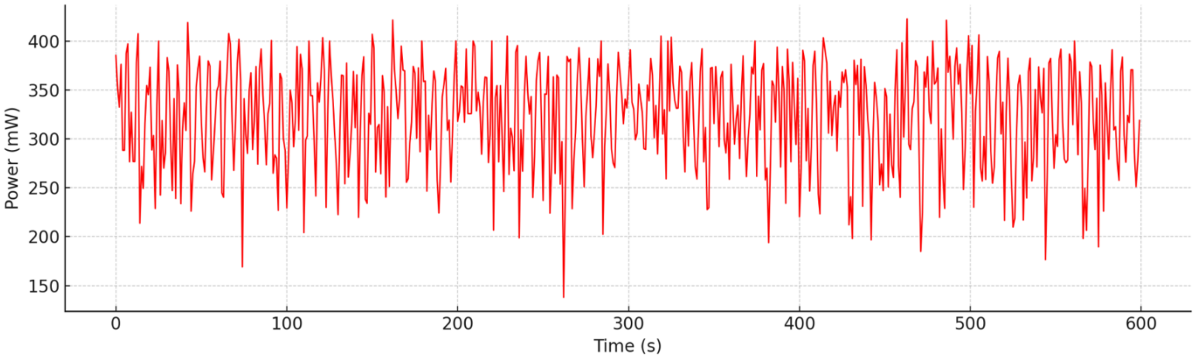


Figure 6: Power consumption during battery-drain attack.

The graph in Figure 6, on the contrary, demonstrates a chaotic pattern of power consumption typical of a battery-drain attack. Instead of clearly defined sleep phases, there are no periods with a power of 5–10 mW. Instead, power consumption remains constantly at an elevated level, with frequent or continuous peaks of up to 300–400 mW, which last longer than in a normal cycle. Also, retention at a level of 80–100 mW without dips into sleep is recorded - this may indicate hidden background activities or forced keeping of the node in an active state. Peaks occur every 5–6 seconds, which is twice as often as in normal mode. Such a pattern is a clear indicator of an anomaly, as it creates a significant load on the battery and reduces the device's battery life.

To summarize the results, the profile buffer entries were classified according to normal/abnormal activity and standard performance metrics were calculated: precision, recall, percentage of false positives (FP) and false negatives (FN). The results obtained are presented in Table 2.

Table 2
Evaluation of experimental results by performance metrics

Metrics	Value (%)
Precision	93.8
Recall	91.2
False Positives	4.1
False Negatives	3.6

Additionally, the impact of the developed module on the node resources was assessed. On average, the MCU processor load increased by 6–8%, and the RAM usage by 9–12% (with a buffer depth of

60 records and sampling once per second). Under conditions of battery power with a capacity of 1500 mAh, the reduction in battery life was ≈ 3.5%, which is acceptable for most practical scenarios.

Despite the high accuracy and low load, a number of limitations were also identified. The method is less effective for short-term attacks that do not have a pronounced impact on the cumulative energy consumption. In addition, temperature drift can affect the accuracy of measurements, especially when using external sensors. To reduce the error, it is necessary to perform preliminary calibration and take into account the permissible temperature ranges for sensors and resistors.

Overall, the experimental results confirm the practical feasibility of using energy profile analysis as a lightweight anomaly detection mechanism for energy-constrained devices in smart grids.

6. Discussion

The results obtained confirm the assumption that energy profiles can serve as a reliable source of information for detecting anomalous activity in power-constrained devices. Unlike classical IDS systems, which mostly depend on traffic analysis or the use of powerful machine learning models, the proposed approach does not require processing a large amount of data, while providing sufficient accuracy for most real threat scenarios. Building a normal profile and observing its deviation over time allows responding to a wide range of attacks - from battery-drain to forced wake-ups and third-party commands.

There The advantage of the approach is its non-load-bearing nature. The detection system can be implemented directly on the node, which preserves the autonomy of the device and eliminates the need for centralized monitoring. In the conditions of smart networks with a large number of low-power nodes (for example, in industrial or ecological WSN systems), this creates the prerequisites for scalable, inexpensive and energy-efficient protection.

The proposed method can serve as a basis for further research in several directions. In particular, the implementation of automatic updating of the normal profile is promising - for example, using a moving average or exponential smoothing. This will allow adapting to changes in the device's operating modes without losing sensitivity. It is also worth considering integrating the method with the analysis of other side characteristics, such as time patterns of processing or traffic features. Finally, one of the most promising directions is the implementation of distributed anomaly detection in clustered networks, where nodes exchange signals about suspicious deviations without the need for full data exchange.

7. Conclusions

The article proposes a method for detecting anomalous activity in smart networks by analyzing the energy profiles of nodes. The method is aimed at using side characteristics – in particular, power consumption – as indicators of changes in device behavior that potentially indicate violations, compromises or attacks. The main focus was on devices with limited resources, for which classic IDS solutions are too resource-intensive or not adapted to the conditions of limited energy, computing power and the absence of guaranteed permanent connection to the network.

An architecture was implemented that combines an energy consumption measurement module, an energy profile accumulation buffer and a lightweight analysis module with simple rules for detecting deviations. The implementation based on ESP32-S3 and the INA219 sensor demonstrated the ability of the system to function autonomously, without affecting the main tasks of the device. The simulation of typical attacks (frequent wake-up, battery-drain, airwave overload, command injection) demonstrated the sensitivity of the system to changes in the energy consumption profile.

Experimental evaluation showed that the proposed solution achieves high accuracy (93.8%) and sensitivity (91.2%) under real-world load conditions, with a slight increase in microcontroller resource consumption (up to 8% CPU and 12% RAM). At the same time, the proposed approach retains scalability, flexibility, and infrastructure independence - this makes it attractive for deployment in distributed IoT systems, including medical, environmental, agricultural, and industrial applications.

Further research is expected to expand the functionality through adaptive updating of the normal profile, automatic sensor calibration, and the use of a distributed detection model in clustered networks. An additional perspective is opened by combining energy monitoring with other types of collateral monitoring information, such as the temporal structure of traffic or the characteristics of processing phases, which will allow increasing resistance to hidden or short-term attacks. The proposed methodology creates a basis for building lightweight and effective cyber protection mechanisms for a new generation of smart devices, as well as increasing the overall system level of reliability of smart networks due to additional technical diagnostics and self-control mechanisms implemented on its basis.

Acknowledgements

This research was funded by the Ministry of Education and Science of Ukraine under grant 0123U100270.

Declaration on Generative AI

During the preparation of this manuscript, AI-based tools were used to enhance the quality and clarity of the content. Specifically, Grammarly Pro was employed for grammar refinement, and Strike Plagiarism was used for ensuring originality. Additionally, generative AI was utilized for the creation of Figure 1 based on public statistics, as well as for Figures 5 and 6, where the input data was taken directly from the results of the experimental evaluation. The authors manually verified and revised all AI-generated content and accept full responsibility for the final version of this publication.

References

[1] I. Rozlomii, A. Yarmilko, S. Naumenko, Innovative resource-saving security strategies for IoT devices, Journal of Edge Computing 4(1) (2025) 35–56. URL: https://doi.org/10.55056/jec.748.

[2] A. Goudarzi, F. Ghayoor, M. Waseem, S. Fahad, I. Traore, A survey on IoT-enabled smart grids: emerging, applications, challenges, and outlook, Energies 15(19) (2022) 6984. DOI: https://doi.org/10.3390/en15196984.

[3] R. Citroni, F. Mangini, F. Frezza, Efficient integration of ultra-low power techniques and energy harvesting in self-sufficient devices: A comprehensive overview of current progress and future directions, Sensors (Basel, Switzerland) 24(14) (2024) 4471. DOI: https://doi.org/10.3390/s24144471.

[4] E. Faure, I. Rozlomii, A. Yarmilko, S. Naumenko, Protection of IoT networks: cryptographic solutions for cybersecurity management, in: Proceedings of the Third International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2024), Kyiv, Ukraine, 2024, pp. 24-34. URL: https://ceur-ws.org/Vol-3925/paper03.pdf.

[5] I. Rozlomii, A. Yarmilko, S. Naumenko, Resource-efficient solutions for data security at the network level of the Medical Internet of Things, in: Proceedings of the 6th International Conference on Informatics & Data-Driven Medicine, IDDM'2024, ceur-ws.org, vol. 3892, 2024, pp. 171-182. URL: https://ceur-ws.org/Vol-3892/paper13.pdf.

[6] Z. M. Iqal, A. Selamat, A comprehensive analysis of risk-based access control models for IoT: Balancing security, adaptability, and resource efficiency, in: 2024 IEEE International Conference on Computing (ICOCO), IEEE, 2024, pp. 344-349. DOI: 10.1109/ICOCO62848.2024.10928193.

[7] N. Kumar, Internet of Things (IoT) Statistics: Market & Growth Data, DemandSage, 2025. https://www.demandsage.com/internet-of-things-statistics/.

[8] J. P. A. Yaacoub, H. N. Noura, O. Salman, K. Chahine, Toward Secure Smart Grid Systems: Risks, Threats, Challenges, and Future Directions, Future Internet 17(7) (2025) 318. DOI: 10.3390/fi17070318.

[9] A. Alshehri, M. M. Badr, M. Baza, H. Alshahrani, Deep anomaly detection framework utilizing federated learning for electricity theft zero-day cyberattacks, Sensors 24(10) (2024) 3236. DOI: 10.3390/s24103236.

[10] S. Ahmad, S. M. N. Hasan, M. S. Hossain, R. Uddin, T. Ahmed, A. G. M. B. Mustayen, R. Hazari, M. Hassan, S. Parvez, A. Saha, A Review of Hybrid Renewable and Sustainable Power Supply System: Unit Sizing, Optimization, Control, and Management, Energies 17(23) (2024) 6027. DOI: 10.3390/en17236027.

[11] H. S. Shreenidhi, N. S. Ramaiah, A two-stage deep convolutional model for demand response energy management system in IoT-enabled smart grid, Sustainable Energy, Grids and Networks 30 (2022) 100630. DOI: 10.1016/j.segan.2022.100630.

[12] N. Anand, M. A. Saifulla, R. B. Ponnuru, G. R. Alavalapati, R. Patan, A. H. Gandomi, Securing software defined networks: A comprehensive analysis of approaches, applications, and future strategies against DoS attacks, IEEE Access 11 (2024) 1-43. DOI: 10.1109/ACCESS.2024.3520478.

[13] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, V. C. Gungor, Energy consumption of on-device machine learning models for IoT intrusion detection, Internet of Things 21 (2023) 100670.

[14] S. Jamshidi, K. W. Nafi, A. Nikanjam, F. Khomh, Evaluating machine learning-driven intrusion detection systems in IoT: Performance and energy consumption, Computers & Industrial Engineering 204 (2025) 111103. DOI: 10.48550/arXiv.2504.09634.

[15] A. Heidari, M. A. Jabraeil Jamali, Internet of Things intrusion detection systems: a comprehensive review and future directions, Cluster Computing 26(6) (2023) 3753-3780. DOI: 10.1007/s10586-022-03776-z.

[16] Y. Bai, J. Park, M. Tehranipoor, D. Forte, Real-time instruction-level verification of remote IoT/CPS devices via side channels, Discover Internet of Things 2(1) (2022) 1. DOI: https://doi.org/10.1007/s43926-022-00021-2.

[17] I. Rozlomii, A. Yarmilko, S. Naumenko, Vulnerability modeling in cybersecurity of intelligent infrastructure networks, in: Kazymyr, V., et al. Mathematical Modeling and Simulation of Systems. MODS 2024. Lecture Notes in Networks and Systems, vol 1391, 2025, pp. 234-248. Springer, Cham. URL: https://doi.org/10.1007/978-3-031-90735-7_19.

[18] N. Cameron, ESP32 microcontroller, in: ESP32 Formats and Communication: Application of Communication Protocols with ESP32 Microcontroller, Berkeley, CA: Apress, 2023, pp. 1-54. DOI: 10.1007/978-1-4842-9376-8_1

[19] S. Thapa, S. C. KC, Raspberry Pi and ESP32-Based Smart Sensor Network for IoT Platform Integration and Real-Time Environmental Data Monitoring, Bachelor's Thesis (2023). URL: https://urn.fi/URN:NBN:fi:amk-2023120333739.

[20] R. Tian, Design of Electrical Equipment Monitoring Device Based on STM32F411, in: Proceedings of the 3rd International Conference on Signal Processing, Computer Networks and Communications, 2024, pp. 188-193. DOI: 10.1145/3712335.3712369.

[21] J. Lambert, R. Monahan, K. Casey, Power consumption profiling of a lightweight development board: Sensing with the INA219 and Teensy 4.0 microcontroller, Electronics 10(7) (2021) 775. DOI: https://doi.org/10.3390/electronics10070775.