Enabling Local Language Models for IoT Monitoring Description via Distributed Low-Cost Hardware

Juan F. Gaitán-Guerrero[1(✉)], Jose L. López[1], Carmen Martínez-Cruz[2], David Díaz-Jiménez[1], and Macarena Espinilla[1]

[1] Department of Computer Science, University of Jaén, Jaén, Spain
{jgaitan,llopez,ddjimene,mestevez}@ujaen.es
[2] Department of Languages and Computer Systems, University of Granada, Granada, Spain
cmcruz@ugr.es

Abstract. As societies face the growing challenge of caring for elderly individuals and people living with chronic conditions, there is an increasing need for intelligent systems that can monitor daily routines while preserving privacy and autonomy. In this context, the convergence of generative Artificial Intelligence and the Internet of Things opens new possibilities for understanding human behavior through natural language. This work proposes a distributed architecture that enables the local execution of lightweight Large Language Models on affordable edge devices to interpret sensor data and generate readable activity summaries. Rather than depending on cloud services, the system follows a conceptual framework designed to scale across real-world homes, adapting to their complexity. A functional prototype was developed and validated using real activity data. Results show that, given the characteristics of the models suitable for small computing units and their extensive use of hardware resources, distributing tasks across devices becomes essential. The proposed architecture contributes to more human-centered monitoring solutions for ambient assisted living environments, while preserving privacy.

Keywords: Edge computing · Large Language Models · Ambient Assisted Living · Internet of Things · Privacy-Preserving Monitoring

1 Introduction

In recent years, societies at large have undergone a profound transformation in the way technology is perceived, utilized, and interacted with. In this context, generative Artificial Intelligence (AI) has emerged as a key driver of change, reshaping multiple and diverse aspects of daily life. Its impact has led to a technological landscape that, until just a few years ago, seemed unimaginable and beyond the reach of most. Beyond its technical capabilities, generative AI has played a significant role in making technology and knowledge about AI more

J. Bravo et al. (Eds.): UCAmI 2025, LNNS 1818, pp. 462–473, 2026.
https://doi.org/10.1007/978-3-032-16992-1_44

accessible to non-experts, thus contributing to the democratization of both access and understanding. Among the most prominent manifestations of this paradigm is the rise of Large Language Models (LLMs), a family of generative AI systems that have redefined how humans communicate with machines and access information [17]. These models enable more natural conversational interfaces and support systems capable of interpreting, generating, and summarizing information in a context-aware manner.

LLMs widespread adoption is evident in platforms such as ChatGPT, which surpassed 400 million active users in early 2025 [14]. LLMs are no longer confined to research settings—they are now embedded across a wide range of domains. Recent industry reports estimate that approximately 67% of organizations worldwide have already integrated LLMs into their workflows [8], with retail and e-commerce leading the adoption curve. This widespread integration reflects not only the flexibility and power of these models, but also their growing role in reshaping how individuals and institutions interact with information and decision-making systems.

In parallel, there has been a growing trend in the utilization of Internet of Things (IoT) based systems, driven by their ability to connect physical and virtual entities through a network of different and diverse sensors and devices [15]. These systems enable the contextualization and retrieval of data related to physical or virtual phenomena, allowing for real-time monitoring, automation, and interaction with the surrounding environment. Such capacities have led to its integration into diverse fields, including urban infrastructure and environmental monitoring [2,7], smart energy distribution [13], data-driven agricultural practices [18], intelligent mobility solutions [12], and remote health monitoring [3].

Among these, the fields of active ageing [5,6] and chronic disease management [11]are considered particularly relevant in the current literature. With the increase in the elderly population, there is a growing need to design environments that promote autonomy, reduce dependence on formal and informal caregivers, and enable the early detection of abnormal situations, as well as the regulation of daily habits to support a more independent lifestyle [5,6]. It is important to note, however, that these needs are not exclusive to older adults: many individuals live with chronic conditions that require continuous monitoring [11], such as diabetes, chronic obstructive pulmonary disease, or various cardiovascular disorders. In such cases, the home becomes a key setting for non-intrusive observation of meaningful behaviours and signals, facilitating not only the prevention of risk situations, but also enabling individuals or patients—as well as their caregivers and healthcare professionals—to track the evolution of a person's condition through their daily activities, and to actively support long-term health and personal empowerment.

Despite the vast potential of IoT systems, most of today's intelligent solutions still depend on cloud services or proprietary servers that must manage security protocols and data handling processes. This reliance introduces several significant barriers, including the need for a constant internet connection, increased response latency, the exposure of personal data to external servers, and the high

energy consumption associated with large-scale remote infrastructure. These factors limit the viability and adoption of such systems in sensitive environments or contexts with technological constraints.

As previously mentioned, the increasing adoption of LLMs across diverse domains—driven by their strong performance in general-purpose tasks—motivates our work. Our primary goal is to advance the generalization of their applicability by enabling autonomous, personalized, and in-home monitoring with minimal user interaction and without the need for external data sharing. To this end, we explore the deployment of LLMs on resource-constrained devices, with a focus on human activity recognition in domestic environments. The key point of our proposal lies in the architectural design, which pretends to distribute computational load and tasks across nodes, making the system scalable according to its complexity.

The document is divided into the following sections. Firstly, a review of other similar proposals can be found in the Sect. 2. Next, Sect. 3 defines the conceptual framework proposed for deploying LLMs in edge computing devices. Subsequently, a validation is carried out with the utilization of different sensors, computing units and LLMs in Sect. 4 and finally, the conclusions and future works are discussed in Sect. 5.

2 Related Works

The increasing demand for personalized and private AI services has driven a notable shift toward deploying Large Language Models on edge devices. This trend aims to reduce reliance on cloud infrastructures, improve latency, preserve user data privacy, and make AI more accessible across resource-constrained environments.

The TinyLLM [10] framework emerges as an innovative solution for deploying language models on edge devices, challenging the trend of increasingly large models. This approach enables training and fine-tuning of much smaller models (30–120 million parameters) that can run locally on edge devices, offering high token throughput and accuracy while addressing latency, connectivity, and privacy issues associated with remote models. Continuing with the evaluation of LLMs at the edge, a comprehensive empirical study [4] on the deployment of LLaMa-2 7B across various edge devices identifies memory and/or compute resource limitations as the main bottlenecks. Results show unacceptably low tokens-per-second performance on low-memory Raspberry Pi devices, while high-end platforms like Jetson AGX Orin achieve significantly higher rates, underscoring the need to enhance memory, bandwidth, and computational units on edge devices.

Pushing optimization further, LLMPi [1] investigates quantization techniques for high-performance, energy-efficient LLM deployment on Raspberry Pi, using k-bit and ternary quantization. Findings show that higher levels of quantization improve performance, with BitNet0.7B standing out for its speed and energy efficiency, though extreme quantization may compromise response accuracy. Additionally, a detailed study [9] on the performance of small language models (SLMs)

on edge devices reveals that the interaction between hardware and SLMs significantly impacts edge AI workloads. While TinyLlama exhibits a smaller memory footprint and low latency, larger models such as LLaMa-3 face critical resource constraints on devices like the Raspberry Pi, leading to increased swap memory usage and higher latency.

Finally, EdgeMoE [16] is presented as an on-device inference engine designed for sparse large language models, addressing the challenge of their massive parameter size on mobile devices. This innovative engine keeps non-expert weights in device memory and loads expert weights from external storage only when activated, employing techniques such as expert bit-width adaptation and prediction-based preloading to significantly reduce memory footprint and accelerate inference.

To sum up, although significant efforts have been made to adapt language models (LLMs and SLMs) for edge environments—optimizing local execution and reducing model size (e.g., TinyLLM, LLMPi, LLaMa-2 7B)—most approaches focus on improving efficiency within a single device. While valuable, this perspective falls short in domains such as indoor positioning or lifestyle monitoring, where the variability in the distribution of rooms in a house and the personal behavior patterns (e.g., sequences of daily activities) make it difficult to generalize from IoT sensor datasets, specially as the complexity of the system increases (i.e., scenarios with the need for monitoring individuals in residences for the elderly). These challenges limit scalability and highlight the need for generative AI approaches not only focused on prediction, but also on natural language description—allowing LLMs to act as observers that support clinical follow-up in a more accessible and human-readable form.

3 Proposed Architecture

This section introduces a conceptual architecture for the distributed generation of human-readable summaries from sensor-driven events across multiple interconnected computing units. The proposal is structured in two parts: (i) a general, technology-agnostic framework; and (ii) a concrete validation setup that instantiates its components. The core objective of this work is to enable the distribution of computational load across lightweight edge units, making it feasible to deploy such systems in large-scale scenarios while ensuring their effectiveness and practical viability.

3.1 Conceptual Framework

The proposed conceptual architecture aims to support the distributed generation of meaningful, natural language summaries from sensor-derived data by leveraging a network of interconnected computing nodes. These nodes collect, process, and interpret events generated in physical environments—such as domestic or clinical settings—where human behavior, health patterns, or environmental

conditions are monitored. The novelty of the approach lies in enabling the distribution of model execution itself, either through partitioned inference across nodes, or by assigning independent models to each node, coordinated through a master node that integrates their outputs.

The architecture adopts a modular and lightweight structure composed of three functional layers (see Fig. 1): **device**, **processing**, and **visualization**. These components operate collaboratively within a local network, supporting both distributed reasoning and low-latency interaction.

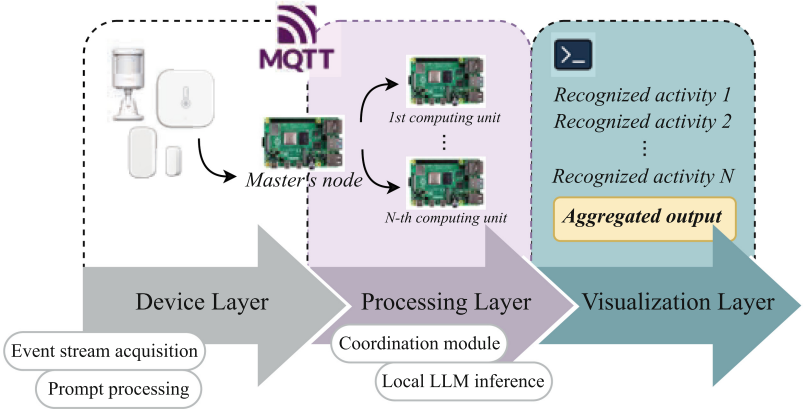

Fig. 1. Architecture of the Proposed Framework.

Device Layer. This layer includes the network of IoT devices responsible for capturing real-time data from the environment. Depending on the application, this may involve motion detectors, ambient sensors or physiological monitors (e.g., continuous glucose monitoring devices), among others. The goal is to collect a continuous stream of raw, heterogeneous data close to the point of origin without relying on cloud infrastructure. This layer communication protocols may vary depending on the different setups of the utilized sensors; whenever possible, sensor nodes may publish data to a local MQTT broker using structured topic hierarchies (e.g., /sensor/motion/kitchen), ensuring efficient communication and enabling scalability as more devices are added. However, certain sensors may require alternative data acquisition methods, such as Bluetooth or access to specialized APIs.

Processing Layer. Local computing units—typically low-power edge devices such as Raspberry Pi boards—perform initial processing of sensor data to extract events of interest, such as abnormal patterns or activity sequences. This detection step may occur in parallel across multiple nodes. Additionally, this layer defines how model execution is distributed across the system. Two strategies are implemented:

- **Model partitioning:** A single language model is logically divided into stages that are assigned to different nodes. Each node executes a portion of the model, contributing to the full inference process. This strategy requires model architectures that support modular execution and inter-node communication for sequential reasoning.
- **Model replication:** Each node hosts an independent instance of a lightweight language model and generates local summaries based on the events it observes. A designated master node then collects these partial outputs, integrates them into a cohesive summary, and may further refine or adapt the content to the target user profile if needed. In this case, the prompt is distributed among multiple computing units, with each processing a separate part of the input.

To support communication between computing nodes, the system relies on a local network (e.g., Wi-Fi or Ethernet). MQTT is used to publish and subscribe to event notifications or summary updates using internal topics (e.g., `/node1/events`, `/node2/summary`). When more structured or synchronous interaction is required—such as delegating specific model stages or coordinating aggregation—gRPC is used to implement remote procedure calls. Synchronization is maintained through time-stamped messages, periodic node status broadcasts, and optional heartbeat signals.

This communication strategy allows for dynamic coordination and role assignment, enabling nodes to adapt to computational constraints or changes in network topology without relying on external services or cloud infrastructure.

Visualization Layer. Once textual summaries are generated, they are made accessible to users through local interfaces, such as dashboards, mobile applications, or simple web pages. These interfaces may retrieve summaries through RESTful HTTP endpoints, receive updates via WebSocket connections, or subscribe directly to MQTT topics. All visualization occurs within the local network, ensuring low latency and preserving data privacy. The goal of this layer is not to adapt the content but to verify that summaries generated within the distributed system can be delivered and displayed consistently across devices.

4 Validation and Evaluation of the Proposed Methodology

To validate the proposed architecture, a functional implementation was developed based on the conceptual framework introduced in Sect. 3. The aim was to test the feasibility of local and distributed reasoning over sensor data in constrained environments, particularly using model replication in the processing layer.

A proof-of-concept deployment was set up involving 4 Raspberry Pi units configured to replicate and execute TinyLLaMA-1.1B in quantized GGUF format, executed locally on each node using the `llama.cpp` framework. This implementation follows the model replication strategy outlined in the architecture:

each node independently receives prompts, performs inference, and returns its partial summary to a central master process; socket communication was used to coordinate prompt distribution and output collection.

Evaluation was conducted manually by comparing the generated summaries against the ground truth derived from sensor data. Specifically, the system's ability to describe human activity events—such as sleeping, waking up, cooking or showering—was analyzed. The comparison focused on the coherence, relevance, and correctness of the generated outputs in relation to the observed data, as well as the consistency of integration across multiple nodes.

4.1 Devices

The utilized devices to instantiate the proposed conceptual framework are technically described subsequently.

Motion Sensors. The Aqara Motion Sensor is a compact motion detector based on passive infrared technology that detects human movement by sensing changes in heat within its surroundings. Designed for indoor use, it offers a range of up to 7 m with a wide field of view. It communicates with the data receiver using the Zigbee protocol, which allows for low power consumption and stable communication with other devices in a local mesh network. The main technical specifications are summarized in Table 1.

Table 1. Main specifications of the Aqara Motion Sensor P1 (model RTCGQ11LM).

Property	Specification
Model	RTCGQ11LM
Technology	Passive Infrared (PIR)
Detection Range	Up to 7 meters
Detection Angle	170°
Communication	Zigbee 3.0
Battery Life	Up to 2 years
Dimensions	30 × 30 × 33 mm
Operating Temperature	−10°C to +45°C
Operating Humidity	0–95% RH

Open/Close Sensors. The Aqara Door and Window Sensor is a compact device based on magnetic detection technology, used to monitor the open or closed status of doors and windows. It operates with a maximum detection gap of 10 mm and communicates via the Zigbee protocol, ensuring low power consumption and reliable local connectivity. These sensors are essential for detecting access-related events in real time. The main specifications are presented in Table 2.

Table 2. Main specifications of the Aqara Door and Window Sensor (model MCCGQ11LM).

Property	Specification
Model	MCCGQ11LM
Technology	Magnetic contact
Detection Range	Up to 10 mm
Communication	Zigbee 3.0
Battery Life	Up to 2 years
Dimensions	41 × 22 × 11 mm
Operating Temperature	−10°C to +50°C
Operating Humidity	0–95% RH (non-condensing)

Ambient Sensors. The Aqara Temperature and Humidity Sensor is used to monitor environmental conditions inside the home. It integrates high-precision components capable of measuring temperature, humidity, and atmospheric pressure (the latter one not being considered in our study). The sensor communicates using the Zigbee protocol and is designed for low-power operation. Multiple units can be placed in different rooms to provide real-time contextual information. Key technical specifications are summarized in Table 3.

Table 3. Main specifications of the Aqara Temperature and Humidity Sensor (model WSDCGQ11LM).

Property	Specification
Model	WSDCGQ11LM
Sensing Capabilities	Temperature, humidity, atmospheric pressure
Communication	Zigbee 3.0
Battery Life	Up to 2 years
Dimensions	36 × 36 × 9 mm
Operating Temperature	−20°C to +50°C
Operating Humidity	0–100% RH

Computing Unit. The Raspberry Pi 4 Model B (8 GB) was employed as the main processing unit within the local edge network. Its compact design, low power requirements, and sufficient memory make it suitable for running lightweight language models and managing inter-device communication. This device is used either as a regular node participating in distributed processing, or as a master unit responsible for coordinating model execution and aggregating outputs. The key specifications of the device are listed in Table 4.

To evaluate the model's capacity to interpret structured temporal data, we constructed a prompt based on real sensor activity recorded during a morning

Table 4. Main specifications of the Raspberry Pi 4 Model B (8 GB version).

Property	Specification
Model	Raspberry Pi 4 Model B (8 GB)
Processor	Quad-core Cortex-A72 (ARM v8) 64-bit, 1.5 GHz
RAM	8 GB LPDDR4-3200 SDRAM
Connectivity	Wi-Fi 802.11ac, Bluetooth 5.0, Gigabit Ethernet
USB Ports	2× USB 3.0, 2× USB 2.0
Storage	microSD slot
Power Supply	5V/3A via USB-C
Video Output	2× micro-HDMI (up to 4K)
Dimensions	85.6 × 56.5 mm

routine. The sequence includes waking up, going to the bathroom, and taking a shower, as inferred from motion and environmental changes. A version of the utilized prompt is shown next:

```
messages = [
    {
        "role": "system",
        "content": (
            "You␣are␣an␣expert␣in␣human␣activity␣
                recognition.␣Given␣sensor␣events␣from␣a␣
                smart␣home,␣infer␣the␣person's␣activity.␣
                Choose␣between:␣going␣to␣bed,␣sleeping,␣
                waking␣up,␣showering,␣cooking.␣"
        ),
    },
    {
        "role": "user",
        "content": (
            "06:58␣-␣Bedroom␣Motion:␣activated\n"
            "07:00␣-␣Bedroom␣Door:␣opened\n"
            "07:03␣-␣Bathroom␣Motion:␣activated\n"
            "07:04␣-␣Bathroom␣Temperature:␣21.0␣degrees␣-
                ␣Humidity:␣72%\n"
            "07:16␣-␣Bathroom␣Door:␣closed"
        )
    }
]
```

Listing 1.1. Input prompt example.

During inference, system performance was monitored on one deployment device. Figure 2 displays both CPU and memory usage associated with a single query to the model. A clear saturation of CPU resources can be observed,

with usage reaching 100% for most of the execution, while memory consumption remained stable.

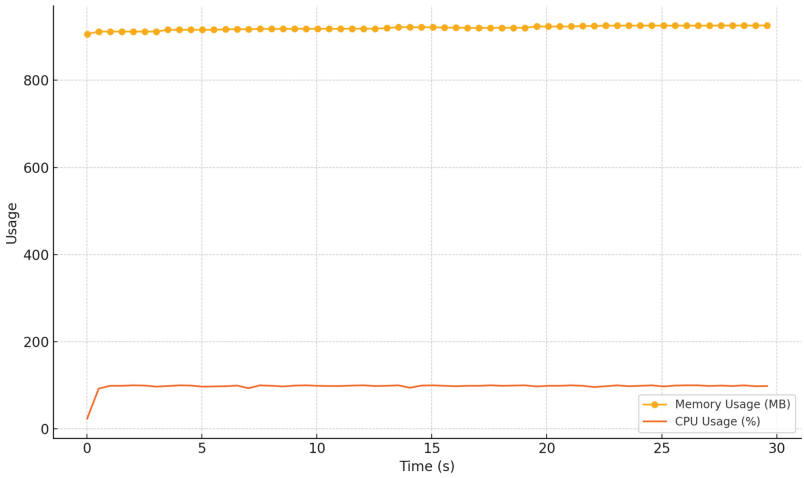

Fig. 2. System Resource Usage During Execution of the Model.

In terms of model accuracy, it was observed that performance degrades as the number of recorded activities in the prompt increases. Additionally, the model tends to misclassify situations where information is incomplete or ambiguous. For example, when a rise in bathroom humidity is detected without any accompanying door events, the model may fail to infer that a shower occurred. Similarly, if sensor updates arrive from multiple disconnected rooms—such as an update in the temperature and humidity in the kitchen immediately after activity in the bathroom—the model struggles to maintain a coherent narrative of the person's actions. These cases highlight the model's reliance on contextual continuity and the presence of supporting signals to make accurate inferences. As a result, these findings reinforce the initial hypothesis that prompts should be broken down into more focused segments. On the contrary, the model performed well in aggregating the summaries of different executions.

5 Conclusions and Future Works

This paper presents a conceptual architecture for the execution of local language models in distributed edge environments, enabling the interpretation of sensor data and the generation of context-aware descriptions without relying on external services. The proposed system was validated through a proof-of-concept implementation using Raspberry Pi devices, showing that local inference is feasible with small models (TinyLLaMA in this case).

Monitoring results confirmed the need for distributing computation due to full CPU saturation during execution, while the observed degradation in inference quality with longer or multi-room prompts—as well as the model's difficulty when handling ambiguous or disjointed events—further validates the distributed prompting strategy

Future work will explore the experimental setup to scenarios with multiple occupants and additional sensors, including indoor localization systems and wearable devices. Another key direction is the development of adaptive user interfaces and interaction mechanisms to facilitate deployment in real-world environments, particularly for non-expert users.

Acknowledgments. This result has been partially supported by grant PID2021-127275OB-I00 and grant PID2021-126363NB-I00 funded by MICIU/AEI/10.13039/5011 00011033, Spain and by "ERDF A way of making Europe", and by grant PDC2023-145863-I00 funded by MICIU/AEI/10.13039/501100011033, Spain and by the "European Union NextGenerationEU/PRTR".

References

1. Ardakani, M., Malekar, J., Zand, R.: LLMPi: optimizing LLMs for high-throughput on raspberry Pi. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 6379–6388 (2025)
2. Buelvas, J., Múnera, D., Tobón V, D.P., Aguirre, J., Gaviria, N.: Data quality in iot-based air quality monitoring systems: a systematic mapping study. Water, Air, Soil Pollut. **234**(4), 248 (2023)
3. Chen, S., Zhao, S., Huang, C.: An automatic malaria disease diagnosis framework integrating blockchain-enabled cloud–edge computing and deep learning. IEEE Internet Things J. 1–1 (2023). https://doi.org/10.1109/JIOT.2023.3304526
4. Dhar, N., Deng, B., Lo, D., Wu, X., Zhao, L., Suo, K.: An empirical analysis and resource footprint study of deploying large language models on edge devices. In: Proceedings of the 2024 ACM Southeast Conference, pp. 69–76 (2024)
5. Díaz-Jiménez, D., López, J.L., Gaitán-Guerrero, J.F., Espinilla, M.: Platera: A comprehensive web-based system and sensor dataset for activity recognition in households environments. In: International Conference on Ubiquitous Computing and Ambient Intelligence, pp. 591–602. Springer, Cham (2024)
6. Díaz Jiménez, D., López Ruiz, J.L., González Lama, J., Espinilla, M.: Methodology based on linguistic protoforms for activity detection in patients with type 2 diabetes mellitus. In: Hawaii International Conference on System Sciences, pp. 1764–1773 (2024)
7. Hossein Motlagh, N., et al.: Unmanned aerial vehicles for air pollution monitoring: a survey. IEEE Internet Things J. (2023). https://doi.org/10.1109/JIOT.2023.3290508
8. Hostinger Tutorials: 67+ key large language model (llm) statistics in 2025 (2025). https://www.hostinger.com/tutorials/llm-statistics. Accessed 18 July 2025
9. Islam, M.R., Dhar, N., Deng, B., Nguyen, T.N., He, S., Suo, K.: Characterizing and understanding the performance of small language models on edge devices. In: 2024 IEEE International Performance, Computing, and Communications Conference (IPCCC), pp. 1–10. IEEE (2024)

10. Kandala, S.V., Medaranga, P., Varshney, A.: Tinyllm: a framework for training and deploying language models at the edge computers. arXiv preprint arXiv:2412.15304 (2024)
11. Lopez Ruiz, J.L., Gaitan-Guerrero, J.F., Martinez-Cruz, C., Díaz-Jimenez, D., Gonzalez-Lama, J., Espinilla, M.: IoT-driven real-time glucose monitoring: empowering diabetes care and prevention. In: International Conference on Ubiquitous Computing and Ambient Intelligence, pp. 119–130. Springer, Cham (2023)
12. Maguluri, L.P., Ananth, J., Hariram, S., Geetha, C., Bhaskar, A., Boopathi, S.: Smart vehicle-emissions monitoring system using internet of things (IoT). In: Handbook of Research on Safe Disposal Methods of Municipal Solid Wastes for a Sustainable Environment, pp. 191–211. IGI Global (2023)
13. Rani, D.P., Suresh, D., Kapula, P.R., Akram, C.M., Hemalatha, N., Soni, P.K.: IoT based smart solar energy monitoring systems. Mater. Today: Proc. **80**, 3540–3545 (2023)
14. Rooney, K.: Openai tops 400 million users despite deepseek's emergence (2025). https://www.cnbc.com/2025/02/20/openai-tops-400-million-users-despite-deepseeks-emergence.html. Accessed 09 Apr 2025
15. Rose, K., Eldridge, S., Chapin, L.: The internet of things: an overview. Internet Soc. (ISOC) **80**, 1–50 (2015)
16. Yi, R., Guo, L., Wei, S., Zhou, A., Wang, S., Xu, M.: Edgemoe: empowering sparse large language models on mobile devices. IEEE Trans. Mob. Comput. (2025)
17. Zhao, W.X., et al.: A survey of large language models (2023). https://doi.org/10.48550/ARXIV.2303.18223
18. Zheng, Q., et al.: Smart contract-based agricultural service platform for drone plant protection operation optimization. IEEE Internet Things J. (2023). https://doi.org/10.1109/JIOT.2023.3288870