Distributed Computing with Language Models at the Edge: A Framework and Prototype

Dinesh Kumar Karthikeyan*
dinesh.k.karthikeyan@jyu.fi
University of Jyväskylä
Jyväskylä, Finland

Roopesh Kumar Shanmugasundaram
roopesh.k.shanmugasundaram@jyu.fi
University of Jyväskylä
Jyväskylä, Finland

Anna-Sofia Paavonen
anna-sofia.s.paavonen@jyu.fi
University of Jyväskylä
Jyväskylä, Finland

Tommi Mikkonen
tommi.j.mikkonen@jyu.fi
University of Jyväskylä
Jyväskylä, Finland

Niko Mäkitalo
niko.k.makitalo@jyu.fi
University of Jyväskylä
Jyväskylä, Finland

Ronny Seiger
ronny.seiger@unisg.ch
University of St.Gallen
St.Gallen, Switzerland

ABSTRACT

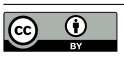
The evolution of cyber-physical systems (CPS) necessitates new methods to manage growing complexity and dynamic interactions between physical and digital domains. This paper presents the Edge Multimodal Intelligence Network on Devices (EdgeMIND) – a cognitive edge computing framework to support CPS development. EdgeMIND deploys I/O nodes powered by large language models (LLMs) across heterogeneous edge resources to process multimodal data streams. These nodes perform runtime decision-making, enabling context-aware interactions with minimal cloud reliance. The framework generates adaptive outputs by integrating retrieval-augmented generation (RAG) with Situational Awareness (SA) enhanced by topic modeling. Empirical evaluations with three LLMs on edge devices show that SA achieves up to 56% latency reduction in CPU-bound nodes and 50% in GPU-based nodes. It also reduces CPU/GPU utilization, memory usage, and thermal load. These results confirm EdgeMIND's effectiveness for context-aware, resource-efficient multimodal processing in CPS as demonstrated in a prototype application. EdgeMIND advances the design of intelligent, efficient edge systems for responsive AI-driven services.

KEYWORDS

Large Language Models (LLMs), Edge AI, Cyber-Physical Systems (CPS), Internet of Things (IoT), Edge Computing, Retrieval-Augmented Generation (RAG)

ACM Reference Format:
Dinesh Kumar Karthikeyan, Roopesh Kumar Shanmugasundaram, Anna-Sofia Paavonen, Tommi Mikkonen, Niko Mäkitalo, and Ronny Seiger. 2025. Distributed Computing with Language Models at the Edge: A Framework and Prototype. In *The 15th International Conference on the Internet of Things (IOT 2025), November 18–21, 2025, Vienna, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3770501.3770511

*Corresponding author.



IOT 2025, November 18–21, 2025, Vienna, Austria

ACM ISBN 979-8-4007-1595-2/25/11.
https://doi.org/10.1145/3770501.3770511

1 INTRODUCTION

Modern software systems are increasingly expansive and distributed, comprising numerous interconnected real-world devices. These systems underpin advanced functionalities in smart grids, homes, cities, and transportation [17]. Operating in highly dynamic environments, they challenge traditional assumptions of design-time stability. This variability increases complexity and can result in emergent behavior that negatively impacts performance and reliability [23]. A key contributor to this complexity is *uncertainty*, which may arise at any stage of system development and propagate through runtime [13]. If not properly addressed, uncertainty can lead to component failures or unpredictable behavior. Managing uncertainty is vital for building reliable, trustworthy systems.

One promising direction lies in the design and deployment of Cyber-Physical Systems (CPS), which tightly integrate computational processes with physical environments. CPS can promote interoperability, support component discovery, and enhance integration, thereby improving efficiency and economic viability [11]. To meet the better performance, scalability, and reliability demands of these systems, *edge computing* has emerged as a critical enabler. By moving computation closer to the data source, edge computing reduces latency, improves responsiveness, and enables decentralized intelligence–key capabilities for next-generation CPS applications. With the advent of the Artificial Intelligence of Things (AIoT) [9], CPS must now also contend with massive, multimodal, and heterogeneous data streams generated at high velocity. These data streams–ranging from images and text to time-series sensor data–are diverse in content, format, and reliability.

To extract timely and actionable insights, modern CPS must adopt multimodal processing architectures capable of handling both real-time (streaming) and delayed (batch) data simultaneously. Edge computing supports this need by enabling low-latency, on-device processing of high-frequency, heterogeneous data, including images, text, and sensor readings.Future networks must support context-aware, human-centric computing to sense, interpret, and adapt system behavior based on dynamic environmental or system states. For example, a context-aware traffic management system may dynamically adjust signal timings based on current congestion, incidents, or weather conditions [6]. These adaptive capabilities

are essential for intelligent and responsive CPS in domains such as smart cities, healthcare, and transportation.

Despite ongoing advancements, current CPS frameworks struggle with scalability, multimodal data integration, and near real-time adaptability. The growing diversity of IoT devices further complicates seamless integration within smart environments. These challenges arise both from the characteristics outlined earlier and those summarized in Table 1, which presents a range of CPS domains alongside their enabling technologies, key challenges, and deployment restrictions [2, 5, 18, 27, 29, 32]. While not all enablers explicitly mention edge computing or AI, many of the challenges (e.g., latency sensitivity, data privacy, real-time processing, and infrastructure constraints) highlight the critical need for decentralized, context-aware computation empowered by Edge AI. Edge computing, combined with AI, including large language models (LLMs), enables intelligence closer to the sources. This reduces communication, preserves data locality, and improves responsiveness, which are essential properties of next-generation CPS.

Motivated by these needs, we propose the **EdgeMIND** framework. EdgeMIND leverages edge computing, multimodal data processing, and context-aware AI mechanisms—including LLMs—to dynamically optimize resource allocation and support near real-time decision-making. This enables scalable, interoperable, and responsive CPS applications across diverse smart domains.

This paper's key contributions include:

- **EdgeMIND:** A modular, cognitive edge computing architecture integrating multimodal data processing and LLM-based context-aware CPS with reduced cloud dependency.
- **Prompt-Centric Optimization:** Unlike pruning or sharding approaches, EdgeMIND leverages input filtering (RAG and topic modeling) to enable lightweight situational awareness—reducing token load, minimizing hallucinations, and avoiding retraining.
- **Prototype Validation:** Demonstration of EdgeMIND's performance advantages, including reduced latency and resource use, through an intelligent office assistant as a prototypical implementation.

2 BACKGROUND AND RELATED WORK

CPS rely heavily on Internet of Things (IoT) infrastructures for real-time sensing, data transmission, and actuation. This section reviews key IoT architectural layers and recent advancements in edge computing and Edge AI, with a focus on their relevance to latency-sensitive, resource-constrained CPS applications. The IoT integrates sensing technologies, embedded computing, and connectivity to facilitate real-time data acquisition, communication, and processing among distributed devices. According to [14], a widely adopted IoT architecture is structured into three primary layers:

- **Perception Layer:** Handles physical data acquisition and initial signal processing using embedded systems. Devices in this layer often form self-organizing networks to manage high-volume, heterogeneous sensor data.
- **Network Layer:** Transmits collected data using infrastructures (e.g., Internet, mobile, or satellite networks). Ensuring data reliability and optimizing data flow are essential, given the scale and bandwidth constraints of modern IoT systems.
- **Application Layer:** Provides interfaces and domain-specific services (e.g., in smart homes, healthcare, or transportation). Effective implementations depend on accurate sensing, real-time data filtering, and efficient storage solutions.

Building upon these foundations, recent studies have examined the role of Edge Computing (EC) in reducing latency and cloud dependency. For example, [31] proposed a layered edge architecture–incorporating data collection, edge processing, service orchestration, and security–which demonstrated a 59.46% reduction in latency when processing multimodal data compared to traditional cloud-based models. These findings validate the role of layered edge architectures in addressing latency and scalability issues. They provide a foundation for our EdgeMIND framework, which extends these capabilities through AI-driven reasoning and dynamic context-awareness. While many existing approaches focus primarily on improving system performance (e.g., reducing latency or cloud dependency), they often lack intelligent multimodal reasoning and adaptive context-aware decision-making, which are essential for real-time CPS applications in dynamic environments. Our proposed EdgeMIND framework fills this gap by integrating situational awareness, LLM-based decision support, and multimodal data handling to enable intelligent, responsive CPS solutions.

As IoT adoption increases, the volume of generated data poses additional challenges in latency, scalability, and energy efficiency. Conventional cloud-centric approaches often fail to meet the low-latency requirements of CPS due to communication overhead and centralized bottlenecks [4]. Edge AI addresses these limitations by enabling local, AI-driven processing close to the data source. Tasks such as anomaly detection, prediction, and adaptive control can be executed in near real-time without cloud round-trips, reducing latency, enhancing privacy, and saving bandwidth and energy by filtering out redundant transmissions [10]. Nonetheless, Edge AI must also contend with the limited computational power, memory, and energy available on edge devices. To balance intelligent processing with these constraints, researchers have proposed strategies such as model compression, feature/data quantization, and adaptive sampling or transmission. These techniques streamline AI models and dataflows to improve performance and energy efficiency–crucial for deploying AI in resource-constrained edge environments [20].

Most IoT edge devices are built on resource-efficient System-on-Chip (SoC) platforms, making them suitable for lightweight preprocessing and local workflows [22]. For heavier workloads, data is often offloaded to servers or cloud infrastructures, forming a computing continuum. This hierarchical architecture supports both real-time responsiveness at the edge and scalable batch processing in the cloud, offering a flexible and robust solution for CPS.

Prior research has shown the effectiveness of IoT and edge computing in CPS, but deeper integration of AI reasoning, situational awareness, and multimodal decision-making remains limited. EfficientLLM [25] uses pruning during pretraining for edge-friendly LLMs, EdgeShard[30] shards models across edge and cloud to improve latency, and Edge-LLM[28] applies layer-wise compression with adaptive tuning. EntroLLM[21] leverages mixed quantization and entropy encoding, while Edge-PRUNE[3] demonstrates collaborative inference for improved latency and energy. Less explored are approaches that optimize how the model is queried

Table 1: Edge Computing Use Cases and Challenges across CPS Domains

Domain	Enablers	Challenges	Restrictions
Smart Office / Classroom (Work & Learning Environments)	IoT for real-time data collection, AI-driven analytics, Integrated Workplace Management Systems, automated control (light, climate).	Complex device integration, flexible environment requirements, cybersecurity risks, high implementation costs.	High costs, network dependency, interoperability issues, infrastructure limitations, data privacy concerns.
Smart Home	Edge computing, data compression, predictive algorithms, adaptive filtering, dual prediction schemes.	Managing IoT data, high energy consumption, latency, communication overhead, integration costs.	Edge device limitations, prediction errors, high integration costs, data privacy concerns.
Smart Healthcare	IoT devices, predictive AI, telemedicine.	Real-time healthcare data integration, privacy-critical processing, compatibility with healthcare IT.	Regulatory constraints, data privacy concerns, high infrastructure costs, interoperability issues.
Smart Manufacturing	Edge computing, IoT integration, predictive maintenance, cybersecurity solutions, energy optimization.	Complex real-time data, cybersecurity risks, legacy system integration, energy inefficiency.	Edge device limitations, storage constraints, cybersecurity vulnerabilities, integration difficulties.
Smart Agriculture	IoT sensors, machine learning, robotics.	Digitalization gaps, device compatibility issues, high deployment costs.	Data privacy concerns, high costs, need for skilled labor.
Hazard-Prone Areas	IoT devices, satellite tech, predictive analytics.	Unstable connectivity, data accuracy, coordination in emergencies.	Connectivity vulnerability, infrastructure fragility, high costs, specialized skills required.
Smart City	IoT networks, cloud-edge systems, mobile apps.	Citizen participation, cyber threats, data privacy challenges.	Infrastructure costs, privacy regulations, interoperability challenges, dependency on policy frameworks.

(prompt handling, filtering, RAG) rather than restructuring or compressing it. EdgeMIND addresses this gap via a prompt-centric, context-aware filtering layer that reduces token usage, mitigates hallucinations, and maintains responsiveness under resource constraints—emphasizing dynamic, lightweight decision logic instead of structural model modifications.

3 EDGEMIND FRAMEWORK

The EdgeMIND framework integrates IoT, lightweight edge computing, and emerging language model capabilities to support intelligent, resource-aware decision-making in CPS. It addresses key challenges such as data heterogeneity, latency sensitivity, and incomplete or noisy context, all of which are common in low-resource, real-time environments. While conventional AI methods rely on rigid rules or task-specific models, large language models (LLMs) offer flexible reasoning and natural language interaction—when appropriately constrained for edge scenarios. Instead of deploying full-scale LLMs directly on edge devices, EdgeMIND employs a modular, hierarchical (layered) architecture. Lightweight local components provide immediate filtering and response, while more resource-intensive tasks (e.g., vector search, retrieval, and complex reasoning) are offloaded to more capable nodes in the edge network.

Recognizing that LLMs are computationally demanding, EdgeMIND does not seek to apply them indiscriminately. Instead, it explores when and how scaled-down or orchestrated LLM components can enhance tasks such as context-aware reasoning, semantic abstraction, and multimodal interpretation. Techniques like quantization, knowledge distillation, and prompt optimization help to enable this balance. EdgeMIND investigates how LLM capabilities can be harnessed, enabling adaptive, transparent, and interactive edge intelligence within CPS constraints. Figure 1 illustrates the EdgeMIND, which comprises the following core components:

(1) **Multimodal Input/Output Modules**: EdgeMIND ingests diverse data types including streaming sensor data, media files and structured/unstructured text from the specific CPS domains. Inputs are normalized into a unified structure to enable consistent downstream processing [1].
(2) **Input Processing**: A lightweight preprocessing pipeline performs data chunking, modality tagging, and metadata annotation. This stage ensures that input data is semantically structured and optimized for embedding and retrieval.
(3) **Situational Awareness**: To reduce irrelevant context and improve inference precision, EdgeMIND employs a training-free topic modeling approach [7] that extracts dominant themes from user prompts and input data. These topic vectors are used to guide context filtering and chunk selection, helping to avoid performance degradation from excessive or noisy prompt content [12].
(4) **Data Storage and Retrieval**: Embedded representations of input data are stored in a vector database, enabling efficient semantic retrieval based on similarity and topic relevance. This supports a RAG pipeline that grounds the LLM's outputs in factual, domain-relevant content.
(5) **Contextual Prompt Construction and Output Processing**: EdgeMIND combines retrieved data, domain-specific instructions, and user queries into a structured prompt format to guide the LLMs. To enhance multi-step reasoning and generate accurate, actionable outputs, the framework incorporates advanced prompting strategies such as Chain-of-Thought (CoT) and ReAct (Reasoning and Acting) [23],

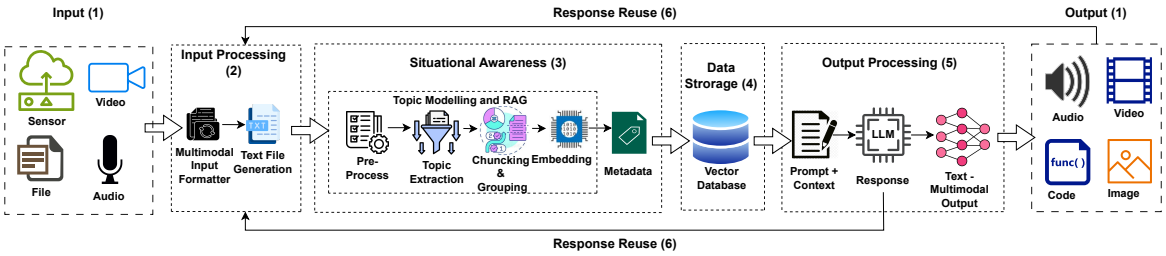


Figure 1: EdgeMIND Framework. This framework processes multimodal inputs followed by situational awareness, where topic modeling is performed with RAG and a vector database, generating multimodal outputs.

which enable interactive decision-making within resource-constrained edge environments, supporting responses delivered through the input/output interface at runtime.

(6) **Response Reuse and Efficiency**: To support stateful, multi-turn interactions, generated outputs can be looped back into the context for subsequent queries [26]. This improves dialogue continuity and reduces redundant processing, enabling the system to accumulate contextual knowledge.

4 INTELLIGENT OFFICE ASSISTANT

Building on the EdgeMIND framework and addressing the challenges summarized in Table 1, we present an instance of the framework in the form of an intelligent office assistant (EdgeMIND-IOA). With the rise of AI and IoT, modern offices increasingly aim to automate tasks such as meeting summarization, environmental monitoring, and personalized insight generation. However, most intelligent office assistants rely on cloud-centric architectures, transmitting sensitive voice, visual, or sensor data to remote servers for processing. This introduces latency, consumes bandwidth, and raises significant privacy concerns—especially in business settings that demand secure and adaptive behavior [16]. Additionally, these systems tend to be static and task-specific, with limited adaptability to dynamic office conditions. Figure 2 presents the prototype of the EdgeMIND-IOA, illustrating its local-first design that reduces dependence on cloud services and enables secure, low-latency reasoning at the edge. Figure 3 shows the AI-generated response to user queries, contextualized based on the whiteboard sketch provided earlier. The key functionalities of EdgeMIND-IOA include:

Multimodal Input Preprocessing: EdgeMIND-IOA ingests and standardizes heterogeneous input types from office environments into a unified textual format for downstream processing.

- **PDF-to-text extraction** is done using LlamaParse, which maintain document structure, extract tabular data, and enhance context by LLM-based augmentation [15].
- **Automatic Speech Recognition (ASR)** is done by *Whisper*, a robust, multilingual speech-to-text model [19].
- **Image-to-text conversion** is enabled by the Phi-3.5-mini.5 Vision-Language model, which interprets visual elements to generate descriptive captions [1].

While LlamaParse is typically associated with cloud-based LLM pipelines, its architecture is highly modular and it supports offline use. In EdgeMIND-IOA, we adapt its document parsing and indexing components to operate directly on edge devices, which allows for PDF-to-text extraction and context-aware structuring without transmitting data to the cloud and thereby prserves privacy and aligns with the edge-first design of the framework. Note that in EdgeMIND-IOA, early-stage data preprocessing–including tasks such as speech transcription, document parsing, and image captioning–is performed directly on the mobile edge device (e.g., smartphone). More compute-intensive tasks (e.g., metadata enrichment, topic clustering, semantic embedding generation, and LLM inference) are offloaded to a nearby local edge node. This balances performance and energy constraints, minimizes latency, and ensures privacy without relying on cloud-based services.

Situational Awareness and Enhanced Data Processing: EdgeMIND-IOA applies topic modeling [7] to enhance its understanding of the office environment and improve contextual retrieval. It is used to automatically identify dominant topics within transcribed meetings, extracted documents, and visual notes, enabling the system to organize and access relevant information more effectively.

- The meeting transcripts are analyzed to extract discussion themes (e.g., project updates, task assignments, issues raised), which are then used to label and cluster content for targeted summarization or Q&A.
- Visual content (e.g., whiteboard sketches or slides) is categorized based on topics to support multimodal recall.
- Documents like agendas or reports are indexed by topic segments to improve focused retrieval during interactions.

This topic-aware organization reduces redundant or irrelevant context in prompts, especially in multi-turn interactions, improving the precision and interpretability of LLM responses. Compared to basic RAG pipelines, this strategy ensures that EdgeMIND-IAO delivers office-specific, concise, and situationally aligned outputs.

Metadata Generation and Vector Database Storage: In EdgeMIND-IOA, multimodal inputs (e.g., meeting transcripts, whiteboard captures, and document excerpts) are processed and enriched with metadata including the extracted text, timestamps, and associated discussion topics (e.g., budget decision, hiring strategy). Semantic embeddings are used to support similarity search, while similarity scores (distance) are computed at query time to rank retrieved items and are not stored as part of the metadata. Topic tagging is performed using lightweight topic modeling techniques suitable for edge deployment. These structured records are then indexed in a FAISS-based vector database, hosted on the nearby

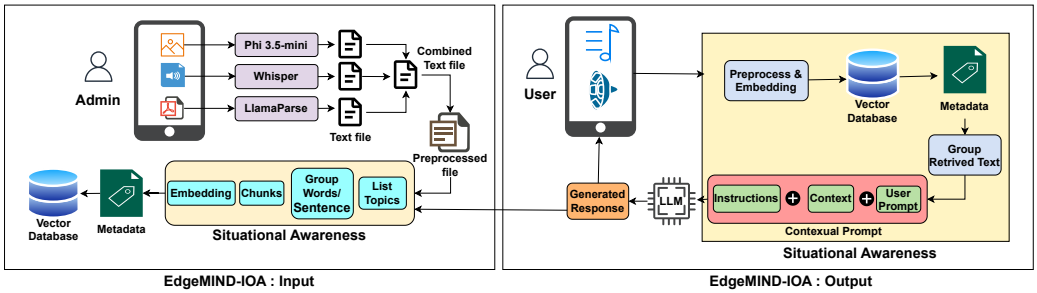


Figure 2: Intelligent Office Assistant (EdgeMIND-IOA) as an instance of EdgeMIND on mobile phones.

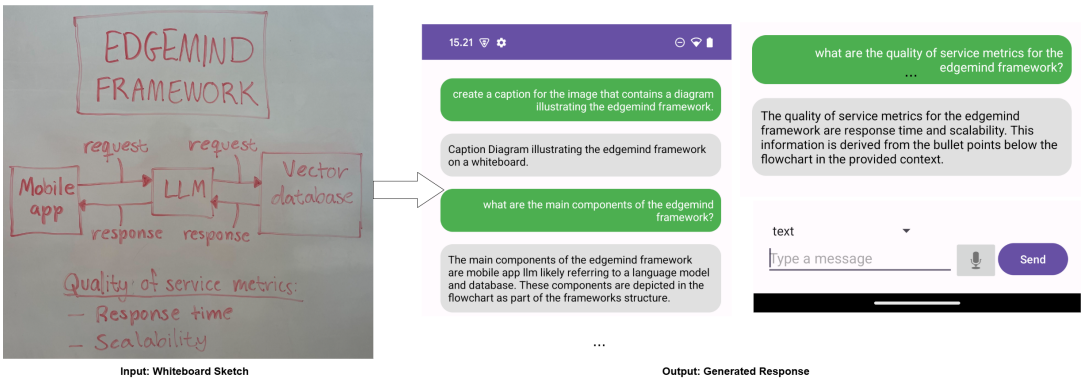


Figure 3: Response generated by the EdgeMIND-based Intelligent Office Assistant for a user prompt on mobile.

local node. The metadata is stored in a structured format to enable efficientsimilarity-based retrieval for follow-up queries such as *"What did we decide about budget last week?"*, while also supporting topic-based categorization of future inputs. As new data arrives, the EdgeMIND-IAO aligns it with existing topic clusters where relevant or creates new topic nodes when necessary. This dynamic organization supports evolving discussions under coherent topical headings, reduces prompt size, and helps prevent retrieval of irrelevant or redundant context during inference.

Contextual Prompt Design and LLM Querying: To realize the framework's prompting approach in practice, we developed a prompt template tailored to the EdgeMIND-IOA. This template ensures that the LLM only uses contextually relevant information from the vector database, aligns responses with the user's query language, and avoids hallucination. The template is shown in Listing 1. This prompt guides the LLM to produce grounded, precise, and context-aware answers, which is critical for handling multimodal data. Incorporating this template allows the EdgeMIND-IOA to maintain consistency, reduce irrelevant or speculative responses, and provide reliable language-aligned support in interactive office scenarios. While EdgeMIND is designed to accommodate advanced prompting strategies (e.g., Chain-of-Thought (CoT) and ReAct [23]) to support multi-step reasoning and interactive decision-making, the implementation of the EdgeMIND-IOA adopts a lightweight, instruction-based prompt format. As an initial stage of deployment, this approach is well-suited for generating grounded and contextually relevant responses from multimodal meeting data while maintaining efficiency on mobile edge devices.

```
prompt_template = """
<|system|>
You are an expert assistant that answers to the user query.
You are given some extracted parts relevant to the user query.
If you don't know the answer, just say "I don't know." Don't
      try to make up an answer.
It is very important that you ALWAYS answer the question in the
      same language the question is in. Remember to always do
      that.
Use only the following pieces of context to answer the question
      at the end.
<|end|>
<|user|>
Context: {context}
Question is below. Remember to answer in the same language:
Question: {question}
<|end|>
<|assistant|>
"""
```

Listing 1: Prompt template used in the EdgeMIND-IOA

Event-Driven Programming and Communication Protocols: In the EdgeMIND-IOA prototype, the EdgeMIND framework is instantiated to manage asynchronous IoT data streams using event-driven Python programming. Lightweight communication protocols (e.g., MQTT and ZeroMQ) enable efficient data exchange between mobile edge devices and local EdgeMIND modules, enhancing responsiveness under varying workloads [8]. Modality-specific data streams–speech, visual input, and sensor signals–are routed to their respective lightweight processing pipelines, as introduced earlier. This scheduling mechanism ensures that processing remains efficient,

Table 2: Prompt Categorization by Complexity Level

Complexity	User Prompt (experiment question)
Simple (factual questions)	What clustering algorithm is used to group similar features extracted by the CNN?
Medium (contextual understanding)	How are the two halves of the diagram related to different DevOps practices, and what does the central element represent?
Advanced (reasoning and synthesis)	What are the main ethical challenges in Machine Learning and Deep Learning highlighted in the context, and why is it important to address them?

scalable, and context-aware across heterogeneous data sources. EdgeMIND's modular implementation allows deployment on mobile devices and edge-grade hardware, offering adaptability across office configurations with varying capacities. This use case illustrates how EdgeMIND facilitates context-aware processing and modular deployment, aligning with the performance, privacy, and adaptability goals of smart office environments.

5 RESULTS AND DISCUSSION

We evaluated the EdgeMIND-IOA prototype–built using the EdgeMIND framework–in a real-world smart office environment. The evaluation focused on resource utilization, response speed, and the relevance of generated responses across typical office scenarios.

5.1 Experiments

All evaluations were conducted on a Lenovo ThinkBook 16p G5 IRX running Ubuntu 24.04.2 LTS. It is equipped with an Intel Core i9-14900HX CPU (32 cores), 32 GiB RAM, a 1 TB SSD, and an NVIDIA GeForce RTX 4060 Laptop GPU with 8 GiB VRAM. This configuration was used for the CPU and GPU-based inference experiments to ensure a fair comparison of latency and resource utilization. To assess practical applicability in mobile-to-edge deployments, the EdgeMIND-IOA prototype was also deployed on a Google Pixel 8a smartphone running Android 15, powered by a Google Tensor G3 chipset with 8 GiB of RAM. The mobile device captured multimodal inputs–including voice and images–and offloaded them to nearby edge nodes for processing, enabling evaluation under realistic, on-the-go conditions. A lightweight mobile application, introduced in Figure 2, served as the front-end interface for initiating queries, viewing outputs, and managing user-specific configurations. While the app itself does not perform heavy computation, it connects to a nearby edge node responsible for processing user prompts and generating contextual responses with low latency.

To evaluate the performance of the EdgeMIND-IOA, we conducted latency experiments across the three LLMs *Phi-3.5-mini*, *LLaMA2-7B*, and *Gemma* in two architectural configurations: RAG-only and Situational Awareness (SA), which combines RAG with topic modeling. Each configuration was tested on CPU and GPU-based edge nodes using three prompt complexity levels. To reflect realistic usage scenarios and assess model robustness, we defined three prompt complexity levels, as shown in Table 2: simple (factual questions), medium (contextual understanding), and advanced (reasoning and synthesis). Based on the multimodal input–including images related to DevOps, whiteboard sketches and a research paper [24]–uploaded and processed by EdgeMIND during a technical discussion at the office meeting, we designed the prompts to generate a contextually relevant response (cf. Table 2).

Table 3: Latency (in seconds) and relative improvement on CPU-based edge nodes across prompt complexity levels. Bold values indicate lower (better) latency under the Situational Awareness (SA) configuration and significant improvements.

Model	Prompt	RAG (s)	SA (s)	Improvement (%)
Phi-3.5-mini	Simple	30	**22**	**27%**
	Medium	24	**20**	**17%**
	Advanced	54	**45**	**17%**
LLaMA2-7B	Simple	9	**4**	**56%**
	Medium	26	**12**	**54%**
	Advanced	27	**10**	**63%**
Gemma	Simple	2	**1**	**50%**
	Medium	4	**3**	**25%**
	Advanced	4	**2**	**50%**

Table 4: Latency (in seconds) and relative improvement on GPU-based edge nodes across prompt complexity levels. Bold values indicate lower (better) latency under the Situational Awareness (SA) configuration and significant improvements.

Model	Prompt	RAG (s)	SA (s)	Improvement (%)
Phi-3.5-mini	Simple	12.00	**10.00**	**17%**
	Medium	12.00	**11.00**	**8%**
	Advanced	26.00	**25.00**	**4%**
LLaMA2-7B	Simple	2.00	**1.00**	**50%**
	Medium	4.00	**3.00**	**25%**
	Advanced	2.00	**2.00**	0%
Gemma	Simple	0.80	**0.50**	**38%**
	Medium	0.41	**0.36**	**12%**
	Advanced	0.83	**0.56**	**33%**

5.2 Results: Response Speed and Latency

Latency measurements across CPU and GPU edge nodes demonstrate that the Situational Awareness (SA) configuration consistently outperforms the RAG-only baseline across all models and prompt complexities. These trends are clearly illustrated in Tables 3 (CPU-based) and 4 (GPU-based), and in Figures 4 and 5.

On CPU-based nodes, LLaMA2-7B shows the most substantial improvement, with latency reductions of up to 56% for simple prompts (from 9 s to 4 s), 54% for medium (26 s to 12 s), and 63% for advanced prompts (27 s to 10 s). Phi-3.5-mini also benefits significantly, achieving up to 33% lower latency—for example, from 30 s to 22 s on simple prompts. Gemma, being lightweight, shows smaller but consistent improvements–50% faster on simple and advanced prompts.

5.3 Results: Resource Utilization

On GPU-based nodes, the improvements are smaller in absolute terms but remain meaningful. For instance, Phi-3.5-mini shows an

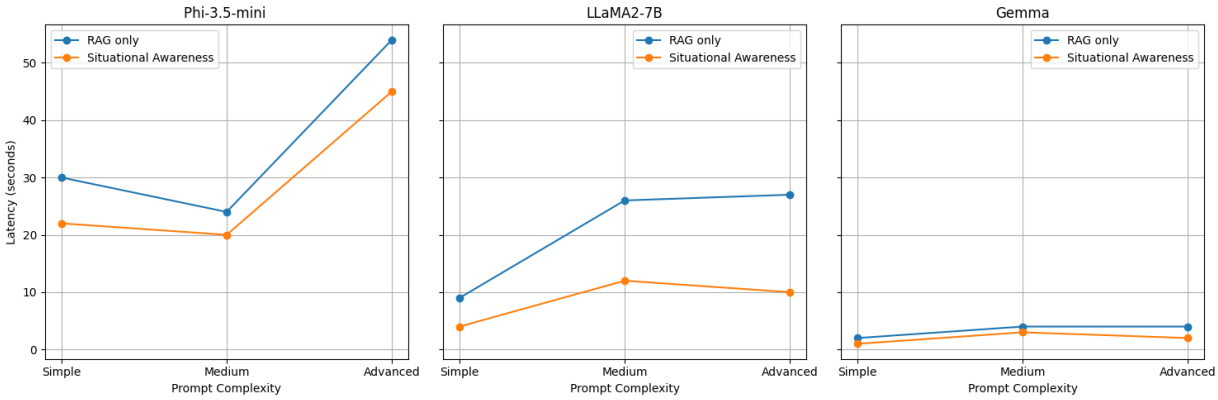


Figure 4: Latency on CPU-based Edge Nodes

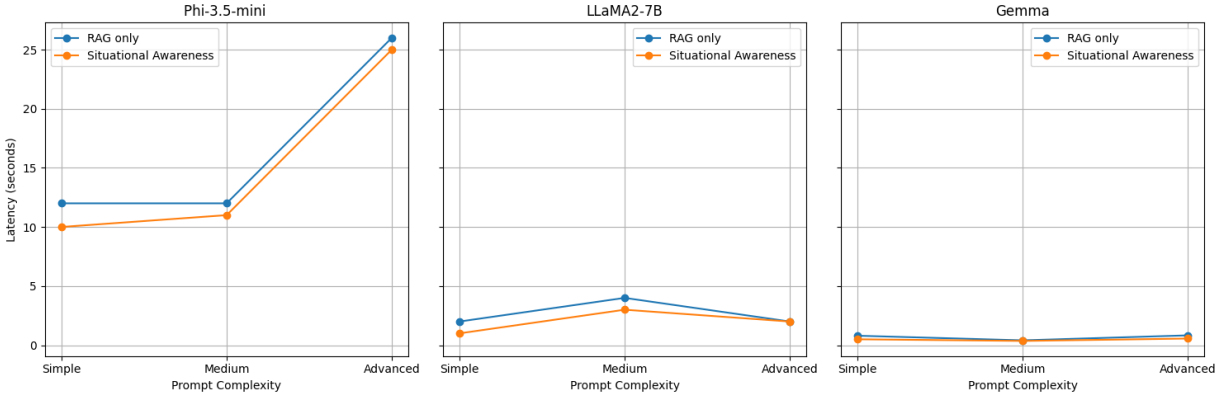


Figure 5: Latency on GPU-based Edge Nodes

8–17% reduction, while LLaMA2-7B achieves up to 50% improvement on simple prompts (2 s to 1 s). Gemma reaches sub-second latency in all cases, with up to 38% improvement (e.g., 0.80 s to 0.50 s). These results confirm that SA enhances contextual accuracy and computational efficiency, especially in CPU-constrained edge environments, making EdgeMIND a practical and scalable solution for real-time, context-aware LLM deployment at the edge.

We monitored system-level metrics during inference: CPU/GPU utilization (%), RAM usage (%), and CPU/GPU temperatures (°C).

CPU-Based Inference Analysis: Table 5 reports CPU resource utilization and temperature across the tested models and prompt types. The SA configuration improved or preserved CPU efficiency compared to the RAG-only baseline. For Phi-3.5, CPU usage decreased from 76.40% to 73.34% under the Simple prompt, with similar reductions for Medium and Advanced prompts. Notably, CPU temperature dropped from 70.30°C to 68.26°C, indicating improved thermal efficiency. RAM usage also remained stable across both modes, with a slight decrease observed in SA. LLaMA2-7B showed higher overall RAM usage (>42%) due to its larger parameter count. However, SA slightly reduced CPU usage and delivered consistent thermal improvements. For instance, under the Medium prompt, CPU temperature dropped from 71.07°C to 66.41°C with SA. Gemma, being a lightweight model, consumed the least RAM (∼ 24%). While CPU usage differences between RAG and SA were marginal, the temperature benefits were more pronounced. Under the Simple prompt, the SA configuration reduced CPU temperature from 66.00°C to 59.67°C, demonstrating its suitability for edge deployment.

GPU-Based Inference Analysis: Table 6 presents GPU resource utilization and temperature data across all test conditions. For Phi-3.5, GPU usage remained consistently high (94–96%), but the SA configuration led to lower GPU temperatures, e.g., decreasing from 50.04°C to 43.62°C (Simple prompt). These reductions indicate more efficient workload management under SA. LLaMA2-7B maintained nearly full GPU saturation in both configurations. Yet, SA consistently reduced GPU temperature, albeit modestly, e.g., from 47.00°C to 46.29°C (Medium prompt). These marginal gains are still important for long-duration inference in embedded or portable systems. Gemma showed the most substantial improvements in thermal and memory performance. Under the Simple prompt, SA reduced GPU temperature from 51.00°C to 44.00°C, and RAM usage dropped from 22.58% to 19.25%. These results reinforce Gemma's efficiency and suitability for resource-constrained edge scenarios. In general, all LLMs generated relevant and correct responses to the queries. Across all tested models and prompt complexities, the Situational

Table 5: CPU Utilization and Temperature (T) for RAG-only and SA across models and prompt types. Bold values highlight lower resource usage or temperature use in SA mode.

Model	Mode	Prompt	CPU (%)	RAM (%)	T (°C)
Phi-3.5	RAG	Simple	76.40	20.90	70.30
		Medium	73.25	19.77	68.09
		Advance	74.95	20.87	71.73
	SA	Simple	**73.34**	**20.88**	**68.26**
		Medium	**72.97**	**19.95**	**70.64**
		Advance	**73.81**	**20.70**	**68.77**
LLaMA2-7B	RAG	Simple	24.22	42.89	62.22
		Medium	24.57	43.54	71.07
		Advance	25.21	43.42	62.03
	SA	Simple	**22.21**	**42.25**	**61.64**
		Medium	**24.25**	**43.40**	**66.41**
		Advance	**23.88**	**42.36**	**63.96**
Gemma	RAG	Simple	25.13	24.13	66.00
		Medium	24.89	24.25	64.83
		Advance	24.27	24.33	65.15
	SA	Simple	**23.97**	**24.00**	**59.67**
		Medium	**24.03**	**23.39**	**60.63**
		Advance	**23.58**	**24.13**	**65.56**

Table 6: GPU Utilization and Temperature (T) for RAG-only and SA across models and prompt types. Bold values indicate lower resource usage or temperature use in SA mode.

Model	Mode	Prompt	GPU (%)	RAM (%)	T (°C)
Phi-3.5	RAG	Simple	95.65	24.28	50.04
		Medium	95.42	25.81	49.72
		Advance	94.49	28.69	64.95
	SA	Simple	**95.25**	**23.69**	**43.62**
		Medium	**94.52**	**25.48**	**45.68**
		Advance	**92.67**	**26.47**	**51.39**
LLaMA2-7B	RAG	Simple	96.00	18.88	45.00
		Medium	97.75	18.91	47.00
		Advance	98.00	19.26	46.60
	SA	Simple	**96.00**	**18.68**	**44.25**
		Medium	**97.13**	**18.83**	**46.29**
		Advance	**97.65**	**19.08**	**46.57**
Gemma	RAG	Simple	95.00	22.58	51.00
		Medium	96.67	23.20	47.33
		Advance	95.60	23.07	47.83
	SA	Simple	**89.00**	**19.25**	**44.00**
		Medium	**94.50**	**19.05**	**47.17**
		Advance	**95.00**	**19.27**	**48.25**

Awareness (SA) configuration improved or preserved computational efficiency and thermal stability, especially in CPU-bound inference scenarios. SA reduced peak temperatures and maintained RAM usage without degrading performance. While all LLMs generated relevant responses, the current evaluation primarily focuses on computational efficiency, latency, and thermal performance. The impact of the Situational Awareness (SA) configuration—particularly the topic modeling layer—on the quality of query responses (e.g., relevance, coherence, or informativeness compared to RAG-only) was not quantitatively assessed. Evaluating this would require additional user studies or benchmark-based metrics, and is therefore considered a limitation of the current work and a direction for future research. These results validate EdgeMIND as a lightweight and effective framework for enhancing runtime efficiency in resource-constrained environments.

6 CONCLUSIONS AND FUTURE WORKS

This paper presents EdgeMIND, an EdgeAI framework integrating multimodal data processing, situational awareness (SA), and retrieval-augmented generation (RAG) to improve LLM inference efficiency on resource-constrained edge devices. Empirical evaluations across three language models and diverse prompt complexities demonstrate that SA consistently reduces latency, computational resource usage, and thermal stress compared to a RAG-only baseline, without compromising inference quality. Deployment on mobile-to-edge architectures validates EdgeMIND's practical applicability in real-world, energy- and resource-constrained environments, advancing the state of the art in context-aware, multimodal edge AI.

Future work will focus on energy-aware orchestration, hardware acceleration, and lightweight model optimization (quantization, pruning, fine-tuning) to enhance efficiency and scalability. We also plan to extend SA through neural topic modeling and dynamic information extraction for improved response relevance, adopt modular containerized designs for flexible and privacy-preserving deployments, and explore EdgeMIND in UAV/aerial networks to support low-latency, distributed inference. These directions aim to evolve EdgeMIND into a robust, scalable, and energy-efficient framework for real-time, context-aware intelligence across diverse CPS domains. It will also consider concrete deployment scenarios across heterogeneous edge platforms, including IoT gateways, on-premise servers, and cloud-edge hybrid systems. Evaluation will be extended to more constrained devices (e.g., Raspberry Pi, Jetson Nano) and additional CPS domains summarized in Table 1 to assess accuracy, robustness, and practical applicability. Moreover, security and privacy aspects—including secure data transmission, anonymization, and access control—will be incorporated to comply with regulatory requirements and ensure safe, trustworthy deployments.

Acknowledgements. This work has been supported by the Business Finland 6G Software project.

REFERENCES

[1] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL] https://arxiv.org/abs/2404.14219

[2] Stephanie B. Baker, Wei Xiang, and Ian Atkinson. 2017. Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access* 5 (2017), 26521–26544. https://doi.org/10.1109/ACCESS.2017.2775180

[3] Jani Boutellier, Bo Tan, and Jari Nurmi. 2022. Edge-PRUNE: Flexible Distributed Deep Learning Inference. arXiv:2204.12947 [cs.DC] https://arxiv.org/abs/2204.12947

[4] Lubomír Bulej, Tomáš Bureš, Adam Filandr, Petr Hnětynka, Iveta Hnětynková, Jan Pacovský, Gabor Sandor, and Ilias Gerostathopoulos. 2021. Managing latency in edge–cloud environment. *Journal of Systems and Software* 172 (2021), 110872. https://doi.org/10.1016/j.jss.2020.110872

[5] Mung Chiang and Tao Zhang. 2016. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet of Things Journal* 3, 6 (2016), 854–864. https://doi.org/10.1109/JIOT.2016.2584538

[6] Seyed Mehran Dibaji, Mohammad Pirani, David Bezalel Flamholz, Anuradha M. Annaswamy, Karl Henrik Johansson, and Aranya Chakrabortty. 2019. A systems and control perspective of CPS security. *Annual Reviews in Control* 47 (2019), 394–411. https://doi.org/10.1016/j.arcontrol.2019.04.011

[7] Mustapha Hankar, Mohammed Kasri, and Abderrahim Beni-Hssane. 2025. A comprehensive overview of topic modeling: Techniques, applications and challenges. *Neurocomputing* 628 (2025), 129638. https://doi.org/10.1016/j.neucom.2025.129638

[8] HiveMQ. 2025. MQTT vs ZeroMQ for IoT. https://www.hivemq.com/blog/mqtt-vs-zeromq-for-iot/ Accessed: March 10, 2025.

[9] Kun Mean Hou, Xunxing Diao, Hongling Shi, Hao Ding, Haiying Zhou, and Christophe de Vaulx. 2023. Trends and challenges in AIoT/IIoT/IoT implementation. *Sensors* 23, 11 (2023), 5074.

[10] Shashikant Ilager, Vincenzo De Maio, Ivan Lujic, and Ivona Brandic. 2023. Data-centric Edge-AI: A Symbolic Representation Use Case. In *2023 IEEE International Conference on Edge Computing and Communications (EDGE)*. 301–308. https://doi.org/10.1109/EDGE60047.2023.00052

[11] Nastaran Jadidi and Mohsen Varmazyar. 2021. *A Survey of Cyber-Physical Systems Applications (2017–2022)*. Springer International Publishing, Cham, 1–29. https://doi.org/10.1007/978-3-030-72322-4_145-1

[12] Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O. Arik. 2024. Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG. arXiv:2410.05983 [cs.CL] https://arxiv.org/abs/2410.05983

[13] Nishanth Laxman, Chee Hung Koo, and Peter Liggesmeyer. 2020. U-Map: A Reference Map for Safe Handling of Runtime Uncertainties. In *Model-Based Safety and Assessment*, Marc Zeller and Kai Höfig (Eds.). Springer International Publishing, Cham, 227–241.

[14] Yijie Li. 2021. Data Processing Flow Analysis of Hierarchical Structure System of Internet of Things. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. 123–126. https://doi.org/10.1109/ICBAIE52039.2021.9389875

[15] LlamaIndex. 2024. LlamaIndex: LLM-powered data framework for unstructured documents. https://www.llamaindex.ai/.

[16] Bala Menon. 2024. The Internet of Intelligence. https://gist.github.com/balamenon/6bb4ffe3b0c6226c75930dce5592127b Accessed: 2025-06-18.

[17] National Institute of Standards and Technology. n.d.. Cyber-Physical Systems. https://www.nist.gov/itl/ssd/cyber-physical-systems Accessed: 2025-02-20.

[18] Sameer Qazi, Bilal A. Khawaja, and Qazi Umar Farooq. 2022. IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends. *IEEE Access* 10 (2022), 21219–21235. https://doi.org/10.1109/ACCESS.2022.3152544

[19] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356 [eess.AS] https://arxiv.org/abs/2212.04356

[20] Jacob Sander, Achraf Cohen, Venkat R. Dasari, Brent Venable, and Brian Jalaian. 2025. On Accelerating Edge AI: Optimizing Resource-Constrained Environments. arXiv:2501.15014 [cs.LG] https://arxiv.org/abs/2501.15014

[21] Arnab Sanyal, Gourav Datta, Prithwish Mukherjee, Sandeep P. Chinchali, and Michael Orshansky. 2025. EntroLLM: Entropy Encoded Weight Compression for Efficient Large Language Model Inference on Edge Devices. arXiv:2505.02380 [cs.LG] https://arxiv.org/abs/2505.02380

[22] Ronny Seiger, Florian Niebling, and Thomas Schlegel. 2014. A distributed execution environment enabling resilient processes for ubiquitous systems. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*. IEEE, 220–223.

[23] Shaghayegh Shajarian, Sajad Khorsandroo, and Mahmoud Abdelsalam. 2024. Poster: Intelligent Network Management: RAG-Enhanced LLMs for Log Analysis, Troubleshooting, and Documentation *(CoNEXT '24)*. Association for Computing Machinery, New York, NY, USA, 27–28. https://doi.org/10.1145/3680121.3699887

[24] Koosha Sharifani and Mahyar Amini. 2023. Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal* 10, 07 (2023), 3897–3904.

[25] Xingrun Xing, Zheng Liu, Shitao Xiao, Boyan Gao, Yiming Liang, Wanpeng Zhang, Haokun Lin, Guoqi Li, and Jiajun Zhang. 2025. EfficientLLM: Scalable Pruning-Aware Pretraining for Architecture-Agnostic Edge Language Models. arXiv:2502.06663 [cs.LG] https://arxiv.org/abs/2502.06663

[26] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations (ICLR)*. https://www.microsoft.com/en-us/research/publication/approximate-nearest-neighbor-negative-contrastive-learning-for-dense-text-retrieval/

[27] Li Da Xu, Wu He, and Shancang Li. 2014. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics* 10, 4 (2014), 2233–2243. https://doi.org/10.1109/TII.2014.2300753

[28] Zhongzhi Yu, Zheng Wang, Yuhan Li, Haoran You, Ruijie Gao, Xiaoya Zhou, Sreenidhi Reedy Bommu, Yang (Katie) Zhao, and Yingyan (Celine) Lin. 2024. Edge-LLM: Enabling Efficient Large Language Model Adaptation on Edge Devices via Layerwise Unified Compression and Adaptive Layer Tuning & Voting. (2024).

[29] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (2014), 22–32. https://doi.org/10.1109/JIOT.2014.2306328

[30] Mingjin Zhang, Xiaoming Shen, Jiannong Cao, Zeyang Cui, and Shan Jiang. 2025. EdgeShard: Efficient LLM Inference via Collaborative Edge Computing. *IEEE Internet of Things Journal* 12, 10 (2025), 13119–13131. https://doi.org/10.1109/JIOT.2024.3524255

[31] Zheng Zhou, Jialin Chen, Bo Jin, Hongwei Wang, Jinhua Zhang, and Jingru Li. 2024. Real-Time Data Processing Strategy Based On Edge Computing In 5G Network. In *2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC)*. 915–919. https://doi.org/10.1109/PEEEC63877.2024.00170

[32] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. 2019. Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. *Proc. IEEE* 107, 8 (2019), 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951