Journal of Edge Computing, 2026, Vol. **5**, Iss. 1, pp. 126–172 10.55056/jec.1054

Revisiting EdgeAI through the lens of communication, storage and computing optimisations

Mateus Roveda, Daniel Lopes Ferreira, Alberth dos Santos Oliveira, Fernanda Schäfer Tesch da Silva, Rafael Kunst, Cristiano André da Costa and Rodrigo da Rosa Righi

Universidade do Vale do Rio dos Sinos, 950 Unisinos Ave., Bairro Cristo Rei, São Leopoldo, 93022-750, Rio Grande do Sul, Brazil

Abstract. Implementing artificial intelligence models on edge devices (EdgeAI) has gained significant popularity due to its potential to enable real-time applications, achieve low latency, and conserve bandwidth. Additionally, reducing dependence on internet connections or cloud infrastructure provides a more secure and reliable execution environment. However, the resource-limited nature of edge devices poses challenges in communication, storage, and computing. Addressing these challenges requires a comprehensive understanding of existing domain optimisation strategies. This survey reviews the current state of the art in EdgeAI optimisation, focusing on communication protocols, storage solutions, and computing architectures that enhance performance and energy efficiency. The contributions of this review are twofold: (i) We highlight key trends, identify gaps in existing research, and propose promising directions for future research to improve the deployment and performance of EdgeAI systems further. (ii) We develop a structured taxonomy that categorises optimisation strategies into computing, storage, communication, and cross-cutting optimisations, offering a clear framework to understand their interrelated approaches and serving as a comparative framework to identify gaps that single-domain surveys often overlook. This survey is a valuable resource for researchers and practitioners seeking to navigate the complex landscape of EdgeAI optimisation and understand the impact of various optimisation pillars and their interactions.

Keywords: EdgeAI, edge computing, artificial intelligence, IoT (Internet of Things), infrastructure optimisation

1. Introduction

With the evolution of Internet of Things (IoT) devices in recent years, artificial intelligence at the edge (EdgeAI) has emerged as a crucial technology capable of performing complex machine learning (ML) tasks directly on peripheral devices such as sensors, cameras, and smartphones. Unlike approaches such as cloud computing, where resources are virtually unlimited but processing is centralised and remote, EdgeAI enables local processing closer to the user, offering faster responses and reducing the need to transfer large amounts of data to remote servers or data centres [62].

The efficient implementation of EdgeAI faces a tripartite set of challenges distinct from CloudAI. First, regarding Communication, edge devices often operate under constrained bandwidth, making the continuous transmission of raw data to the cloud impractical due to high latency and energy costs [61]. Second, Storage limitations limit the deployment of modern deep neural networks (DNNs), which require memory footprints exceeding the capacity of typical embedded devices. Third, Computing resources are bounded by strict power budgets and thermal limits, preventing the direct execution of complex inference tasks [50]. Crucially, these challenges are interconnected: reducing

0000-0003-2479-7627 (D. Lopes Ferreira); 0000-0002-6180-4104 (R. Kunst); 0000-0003-3859-6199 (C. A. da Costa); 0000-0001-5080-7660 (R. da Rosa Righi)
mroveda@edu.unisinos.br (M. Roveda); danielferreira17@edu.unisinos.br (D. Lopes Ferreira); alberth@edu.unisinos.br (A. dos Santos Oliveira); ftesch@edu.unisinos.br (F. S. Tesch da Silva); rafaelkunst@unisinos.br (R. Kunst); cac@unisinos.br (C. A. da Costa); rrrighi@unisinos.br (R. da Rosa Righi)

Received	Accepted	Published	Version of record
2025-07-14	2026-02-13	2026-05-17	2026-05-21

computational load often requires offloading data, which, in turn, strains communication resources. Figure 1 shows a comparison and differentiation between EdgeAI and CloudAI.

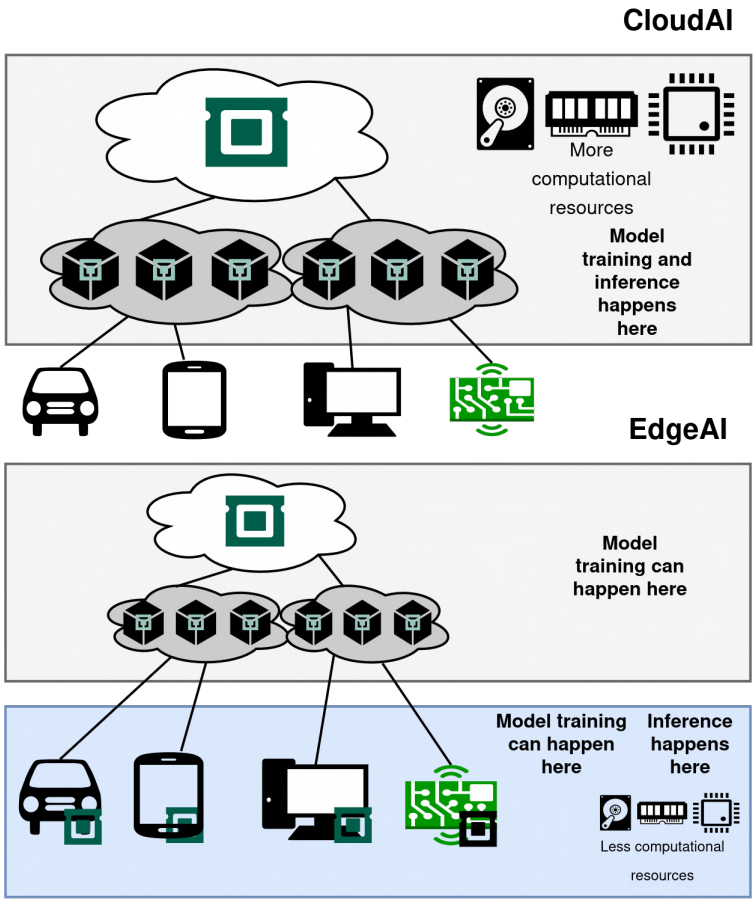

Figure 1: Transitioning from CloudAI to EdgeAI: decentralising inference to edge devices with limited computational resources.

While numerous studies and reviews have explored various aspects of EdgeAI, many focus on specific optimisation domains or need to provide an integrated perspective. For example, Shi et al. [61] highlighted optimisations for communication and energy consumption but left gaps in EdgeAI hardware design. Similarly, Iftikhar et al. [27] conducted a systematic review of resource management using ML and AI methods, but still need to address AI-specific edge optimisations fully. Al Ridhawi et al. [2] provided a comprehensive view of cooperative systems integrating IoT devices with AI and decentralised solutions like Blockchain, but focused primarily on intelligent unmanned aerial vehicles (UAVs), leaving opportunities to explore energy efficiency and broader applicability across various devices and scenarios.

To address these gaps, this survey comprehensively reviews the current state of the art in EdgeAI optimisation, encompassing communication protocols, storage solutions, and computing architectures. We systematically analyse recent advancements and develop a detailed taxonomy that categorises optimisation strategies into computing, storage, communication, and cross-cutting optimisations. This taxonomy provides a structured framework for understanding the interdependencies and synergies within EdgeAI systems, facilitating a holistic analysis of the existing literature.

Our main contributions are:

1. We present a detailed and up-to-date overview of distributed systems architectures and the main computing, storage, and communication technologies used in EdgeAI deployment through an extensive literature review. By analysing optimisation strategies, we highlight key trends,

identify gaps in existing research, and propose promising directions for future research to enhance the deployment and performance of EdgeAI Systems.

2. We propose a structured taxonomy categorising optimisation strategies into computing, storage, communication, and cross-cutting optimisations, enabling a clear understanding of their interrelated approaches.

Our survey is organised into six sections: Section 2 discusses related work focused on communication, storage, or computing optimisations. Section 3 presents the research methodology used in this paper. Section 4 presents our findings on possible optimisations in EdgeAI device infrastructure. Section 5 discusses the current state of the art, introduces our proposed taxonomy, and identifies challenges and open gaps. Section 6 outlines the limitations of this work and suggests directions for future research. Finally, Section 7 summarises the work and presents the main contributions of the research.

2. Related work

This section reviews literature on communication, storage, and general computing optimisations for EdgeAI and similar architectures. The selection process prioritised studies contributing to similar research objectives, identified through academic literature searches using two key tools: Google Scholar and Connected Papers. Google Scholar was pivotal in uncovering surveys addressing comparable research questions, while Connected Papers helped visualise and identify related or derivative works. Following an initial screening, 12 articles were closely analysed. From this set, the 6 most relevant surveys, those strictly focusing on optimisation strategies within the EdgeAI domain rather than general edge computing, were selected and summarised in table 1 to serve as direct benchmarks for this study. These articles are discussed in greater detail in the following paragraphs.

The provided studies collectively offer comprehensive insights into optimisation techniques to enhance the performance and efficiency of AI models in edge computing environments. A significant focus across these works is on model optimisation methods, including hardware optimisation, FL, knowledge distillation, model pruning, and quantisation. Surianarayanan et al. [63] and Boucetta et al. [8] both meticulously review these techniques, highlighting their effectiveness in addressing the resource constraints of edge devices, including satellites in space environments. Liang et al. [39] delve deeper into pruning and quantisation, emphasising their potential to reduce computational complexity and energy consumption while maintaining accuracy in deep neural networks. These methods are crucial for deploying AI models on devices with limited computational resources and power availability.

Another common theme is optimising communication and training processes in distributed and EdgeAI systems. Shi et al. [61] and Duan et al. [16] explore efficient communication techniques and the integration of distributed AI with edge cloud computing, respectively. They identify key challenges such as communication overhead, latency, and network costs, proposing solutions like FL, model compression, and over-the-air computing (AirComp) to enhance efficiency. Khouas et al. [29] focus on methodologies and frameworks that improve the efficiency, privacy, and real-time decision-making capabilities of training ML models directly on edge devices, known as EL. They emphasise the advantages of decentralised computing over traditional cloud-based approaches, addressing data privacy and security challenges in collaborative environments.

While these studies highlight critical technical challenges such as standardisation amid device heterogeneity, privacy assurance, and energy constraints, a significant gap remains in the literature. Existing surveys tend to treat communication, storage, and computing in isolation or focus on only a few pairs of constraints. None offers a holistic analysis that simultaneously accounts for the interdependence among these three pillars. This survey aims to bridge this gap by proposing a unified taxonomy that maps the cross-cutting optimisations required for next-generation EdgeAI systems.

Table 1
Related work: observing how other survey address EdgeAI.

Survey	Main focus	Gaps
Surianarayanan et al. [63]	Reviews various optimisation techniques to enhance the efficiency of artificial intelligence (AI) models for deployment on edge devices.	Difficulty in standardising optimisation techniques due to the heterogeneity of edge devices and the challenge of effectively implementing these methods across diverse edge applications.
Shi et al. [61]	Emphasises the necessity of communication-efficient techniques in EdgeAI to optimise data processing and reduce transmission demands on edge devices.	Handling hardware heterogeneity, ensuring robust privacy and security, and achieving scalability in EdgeAI systems.
Duan et al. [16]	Distributed artificial intelligence (DAI) research powered by edge cloud computing (ECC), highlighting the integration of heterogeneous computing resources to support low-latency, secure and reliable AI services.	High latency and network costs in distributed training, resource constraints on end devices, and unresolved security and privacy threats in the open ECC architecture.
Boucetta et al. [8]	Techniques to optimise AI models for edge-based satellite image processing, addressing the constraints of computational resources and power on satellites.	The challenge of designing hardware suitable for the space environment while maintaining computational efficiency, and the difficulty in obtaining labelled data for training models in federated learning (FL) scenarios.
Liang et al. [39]	It addresses the challenges and optimisation techniques of deep neural networks through pruning and quantisation to improve computational efficiency and reduce energy consumption.	The scalability of pruning and quantisation techniques to larger networks, and their reliance on specialised hardware, remain significant challenges.
Khouas et al. [29]	Comprehensive survey on edge learning (EL), focusing on the optimisation of ML model training at the edge.	Ensuring data privacy and security in collaborative learning, reducing energy consumption and carbon footprint, and training large models at the edge.

3. Methodology

This section describes the research methodology adopted and the procedures used. It highlights the decisions made through a literature review to provide an overview of communication, storage, and computing optimisations in EdgeAI systems. The presented approach identifies technologies, problems and methods used in this field. In addition, the review includes identifying primary studies, applying inclusion and exclusion criteria, and synthesising the results, aiming to provide a comprehensive understanding of the optimisation strategies that can improve the efficiency and effectiveness of EdgeAI systems. The research method was used through the following procedures based on the review of Robben, Englebienne and Kröse [58]:

- Presents the research questions developed for this systematic review.
- Describes the libraries explored and the data collection strategy.

- Defines how the articles were selected, presenting the exclusion criteria and the steps to filter and select the studies.
- Describes the tools and software used to support the research.

3.1. Research questions

Defining the research questions was driven by the observation that existing literature often treats computing, storage, and communication as isolated silos. However, in resource-constrained EdgeAI environments, these resources are deeply coupled (e.g., compression algorithms save bandwidth but consume CPU). Therefore, the following Research Questions were formulated to specifically uncover the trade-offs and synergies between these three pillars, going beyond single-domain optimisations. In order to better identify the main research topics in the selected articles, we divided the questions into two groups: (i) the main questions (MQs), and (ii) the secondary questions (SQs).

The MQs aim to explore the fundamental aspects of technologies and optimisations in EdgeAI systems. Specifically, these questions focus on identifying the computing, storage, and communication technologies used in distributed system architectures and on examining possible optimisations for the data path between the edge, fog, and cloud layers. The following statements define the main MQs addressed in this study:

MQ1: What computing, storage and communication technologies are used in distributed edge AI architectures?

MQ2: What optimisations can be made to the data path for AI workloads between edge-Fog-Cloud?

The SQs explore specific optimisations in EdgeAI systems, including how ML optimisations can improve network throughput, how energy profiling can identify and implement energy-efficiency optimisations, and how high-performance computing (HPC) techniques can increase computing and communication efficiency:

SQ1: What ML optimisations can be made to improve network flow in an EdgeAI system?

SQ2: How can energy profiling be used to identify and implement energy consumption optimisations in EdgeAI systems, considering communication, storage and computing limitations?

SQ3: How can high-performance computing techniques be used to increase computing and communication efficiency in EdgeAI systems?

SQ4: What are the main policies and penalties when addressing quality of service (QoS) in the EdgeAI context?

3.2. Search strategy

Studies were selected from appropriate data sources to answer the research questions, thereby increasing the likelihood of finding pertinent information on communication, storage, and computing optimisations in EdgeAI systems. The search covered the following electronic databases:

- ACM Digital Library (https://dl.acm.org/)
- IEEE Xplore (https://ieeexplore.ieee.org/)
- Elsevier ScienceDirect (https://www.sciencedirect.com/)
- Springer Link (https://link.springer.com/)

In addition, specific keywords were defined to create a search string, which was structured into search units and combined using Boolean operators. To broaden the search scope, acronyms, synonyms, and alternative spellings were also considered. This sequence was applied to the above-mentioned databases, ensuring a comprehensive and detailed search.

```
(EdgeAI OR ‘‘Edge AI’’ OR ‘‘Edge Artificial Intelligence’’ OR ‘‘Edge
                        Intelligence’’)
                              AND
                  (Optimization OR Performance)
                              AND
            (Communication OR Storage OR CPU)
```

The search string covers three main groups: EdgeAI, optimisation, and application area. These groups are combined using the AND operator to ensure that the articles selected cover all these topics. The first group includes terms related to EdgeAI. The second group focuses on keywords associated with optimisation or performance, covering both aspects to cover a wide range of approaches. The third group refers to the specific application areas of optimisation, such as communication, storage and computing, with the terms within each area separated by the OR operator. This search structure allows for a comprehensive exploration of studies relevant to these areas of interest.

3.3. Selection criteria and article selection

We applied the search sequence to the databases presented in Section 3.2 to select the studies. In total, the initial search yielded 4,394 articles: 932 from the ACM Digital Library, 776 from IEEE Xplore, 1,714 from Elsevier ScienceDirect, and 972 from Springer Link. To refine this corpus and select the most relevant research, we employed a fully automated exclusion process comprising seven filters (F1-F7), as detailed in table 2. All considered articles were limited to the English language.

The filtering process was executed using Python scripts developed by the authors to ensure strict adherence to the criteria and reproducibility. Filters F1 through F5 handled metadata cleaning, while Filters F6 and F7 applied semantic checking based on keyword presence in titles, abstracts, and author keywords.

Table 2
Filtering criteria for article selection.

ID	Filtering criterion
F1	Articles published between 2020 and April 2025
F2	Articles containing the search string in the title, abstract, or keyword terms
F3	Eliminate surveys, dissertations, theses, and books
F4	Exclude short papers (4 pages or less)
F5	Eliminate duplicates
F6	Articles covering optimisations in EdgeAI (keyword-based relevance check)
F7	Exclude niche articles (autonomous vehicles, home automation, and healthcare)

During the initial cleaning phase, F1 restricted the interval to publications between 2020 and April 2025. F2 ensured that the base search terms were present in the metadata. F3 and F4 removed non-peer-reviewed or summary works (surveys, theses, short papers), and F5 removed duplicates across databases. Subsequently, F6 was applied to filter articles that specifically addressed optimisation. The script verified if the title, abstract, or keywords contained at least one term from each of the three required groups (optimisation action, metric, and strategy), as defined by the following boolean logic:

```
(Monitoring OR Improve OR Enhance OR Optimize)
```

```
AND
(Latency OR Bandwidth OR Energy OR Cost OR Throughput OR Resource Usage)
AND
(Model Compression OR Quantization OR Pruning OR Load Balancing OR Scaling
OR Edge Offloading OR Resource Allocation)
```

Finally, F7 was applied to exclude niche application domains strictly. The script removed articles containing terms related to health (e.g., healthcare, telemedicine, medical IoT), smart homes (e.g., Home Automation, Smart Appliances), and autonomous vehicles (e.g., UAV, ADAS, V2X, driverless technology). This exclusion was necessary to ensure the survey focuses on generalisable infrastructure optimisations rather than domain-specific constraints, which often rely on vertical solutions that do not apply to the broader edge AI landscape.

Figure 2 illustrates this funnelling process and the percentage of articles removed at each stage.

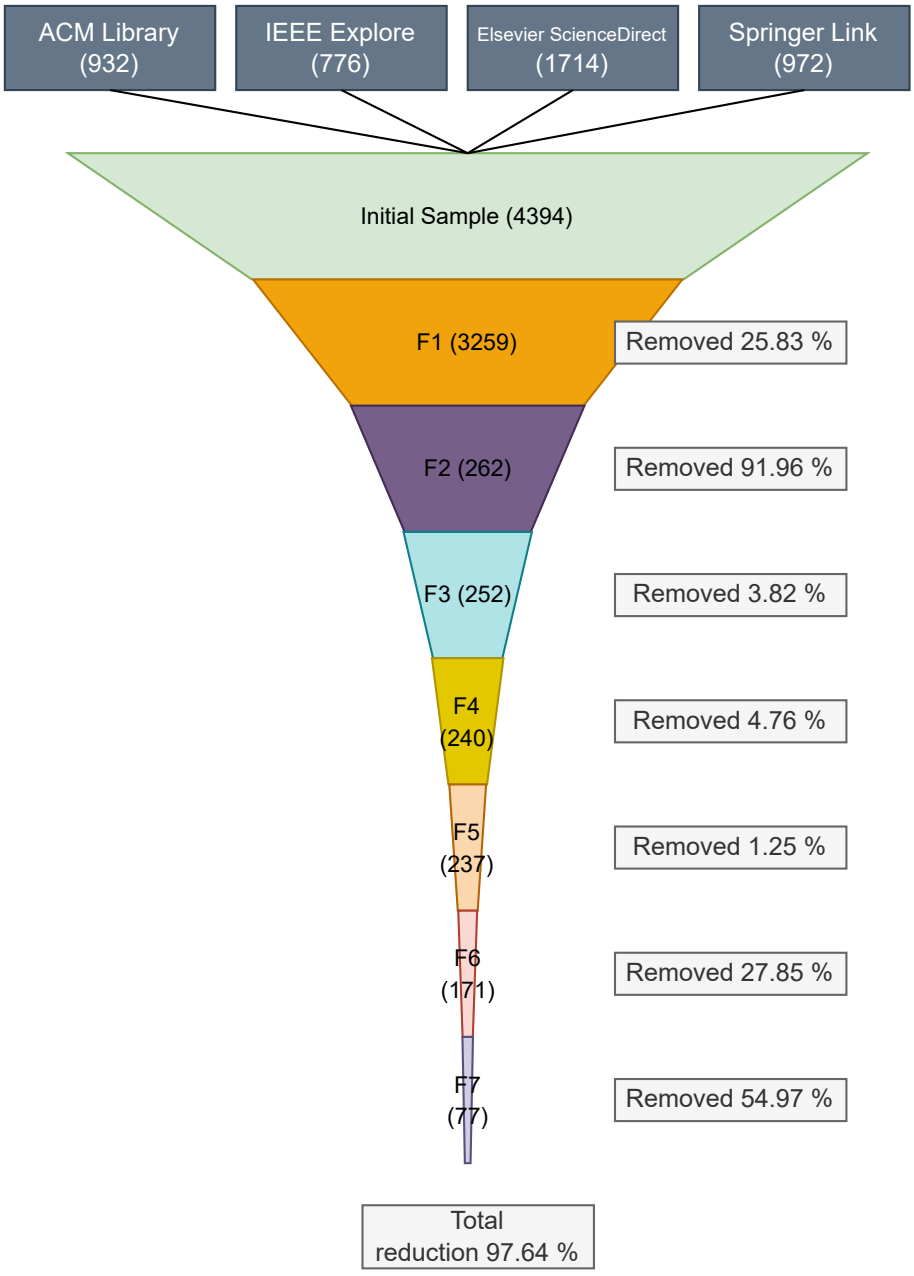

Figure 2: Filtering process for EdgeAI optimisation articles.

These filters resulted in a final selection of 71 articles. It is important to acknowledge that the broad search string was intentionally designed to capture a holistic view of the EdgeAI landscape. While this approach yields a large initial corpus, the subsequent automated filtering steps (specifically the keyword enforcement in F6 and the niche exclusion in F7) were rigorous in selecting studies that represent distinct archetypes of optimisation strategies. Thus, the final 71 articles are considered a representative sample sufficient to construct a comprehensive taxonomy.

3.4. Software and tools

Three main applications were used to assist in the article selection process, as well as Google spreadsheets and CSV files that supported the entire process. Firstly, to help export the results obtained, the Mendeley tool was used, allowing all the articles found through the search string in the databases to be centralized. After this, the selected articles were exported to the Rayyan tool. This second tool helped analyze the articles, allowing exclusions or additions. Finally, the last tool was a script developed by the authors to automate the filtering stages, either by searching for information in the CSV files exported from Mendeley or Rayyan or in the PDF files exported directly from the databases.

For transparency, the authors note that AI-based language tools were used solely for grammatical revision, improvement of English fluency, and minor structural polishing of the manuscript. All scientific content, methodological decisions, analysis, and conclusions remain entirely the responsibility of the authors.

4. Results

The increasing discussion of optimization efforts in EdgeAI is evident in our filtered articles over the years, as shown in figure 3. The number of relevant publications rose from 2020 to 2023, indicating growing interest and research in the field. However, the decrease in the number of articles in 2024 is attributed to the cut-off applied during the execution of this work, which limited the data collection period. Nonetheless, the value for 2024 remains almost the same as in 2021 and 2022, reflecting a consistent interest in the topic. In 2025 the number of articles is smaller due to the analyzed interval, until April.

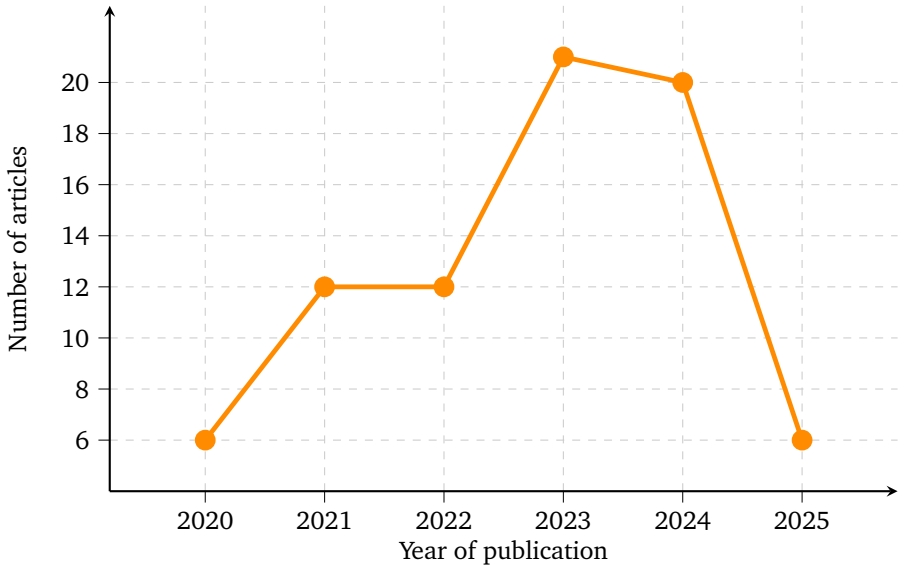

Figure 3: Number of filtered EdgeAI optimization articles by year until April 2025.

The results of the article selection process are summarized in table 3, which provides details on the year of publication, article type, publisher, and the optimization areas covered.

Table 3: Article selection summary.

Year	Article	Publisher	Publication	Optimisation target	
				Hardware	Software
2020	Qiu et al. [56]	IEEE	Journal article	✓	✓
	Hu et al. [22]	IEEE	Conference paper	–	✓
	Xu, Qian and Hu [76]	IEEE	Journal article	–	✓
	Li et al. [33]	IEEE	Journal article	–	✓
	Wang et al. [70]	IEEE	Conference paper	–	✓
	Xia et al. [74]	IEEE	Journal article	–	✓
2021	Li et al. [37]	IEEE	Journal article	–	✓
	Lian et al. [38]	IEEE	Conference paper	–	✓
	Yun et al. [81]	IEEE	Journal article	–	✓
	Chen et al. [13]	IEEE	Journal article	–	✓
	Yang et al. [77]	IEEE	Journal article	–	✓
	Chen et al. [12]	IEEE	Conference paper	–	✓
	Zhou et al. [89]	IEEE	Journal article	–	✓
	Li et al. [32]	IEEE	Conference paper	–	✓
	Lu et al. [46]	IEEE	Journal article	✓	✓
	Minh, Mai and Minh [47]	IEEE	Conference paper	–	✓
	Nunez-Yanez and Hosseinabady [51]	Elsevier	Journal article	✓	–
	Wang et al. [68]	Elsevier	Journal article	–	✓
2022	Yun, Choi and Kim [80]	IEEE	Journal article	–	✓
	Li et al. [35]	IEEE	Journal article	–	✓
	Guo et al. [19]	IEEE	Journal article	–	✓
	Chan et al. [9]	IEEE	Journal article	–	–
	Gong et al. [17]	IEEE	Conference paper	–	✓
	Li et al. [36]	IEEE	Conference paper	–	✓
	Huang et al. [25]	IEEE	Conference paper	–	✓
	Abdel-Basset, Moustafa and Hawash [1]	IEEE	Journal article	–	✓
	Liu et al. [44]	IEEE	Journal article	–	✓
	Desnos et al. [15]	IEEE	Conference paper	–	✓
	Liu et al. [43]	Springer	Journal article	–	✓
	Tuli, Casale and Jennings [67]	Springer	Journal article	–	✓
2023	Xu et al. [75]	IEEE	Journal article	–	✓
	Qu et al. [57]	IEEE	Journal article	–	✓
	Wang et al. [71]	IEEE	Journal article	–	✓
	Binucci et al. [7]	IEEE	Conference paper	✓	✓
	Wen et al. [72]	IEEE	Conference paper	–	✓
	Hlophe, Awoyemi and Maharaj [21]	IEEE	Conference paper	–	✓
	Chen et al. [10]	IEEE	Conference paper	–	✓
	Ji and Qin [28]	IEEE	Conference paper	–	✓
	Zeng et al. [82]	IEEE	Journal article	–	✓
	Zhang et al. [84]	IEEE	Journal article	–	✓
	Psaromanolakis et al. [53]	IEEE	Conference paper	–	✓
	Takeuchi [64]	IEEE	Conference paper	✓	–
	Zhang et al. [83]	IEEE	Conference paper	–	✓
	Lin et al. [41]	IEEE	Conference paper	–	✓
	Li, Bi and Wang [34]	IEEE	Journal article	–	✓
	Liu et al. [45]	IEEE	Conference paper	–	✓
	Gong et al. [18]	IEEE	Conference paper	–	✓
	Askarizadeh, Morsali and Nguyen [4]	IEEE	Journal article	–	✓
	Ang, Rana and Hameed [3]	IEEE	Conference paper	–	–
	Huang et al. [23]	IEEE	Journal article	–	✓
	Yang and Chen [78]	IEEE	Journal article	–	✓

Continued on next page

Table 3 – continued from previous page

Year	Article	Publisher	Publication	Optimisation target	
				Hardware	Software
2024	Ogbogu et al. [52]	ACM	Journal article	–	✓
	Wu et al. [73]	IEEE	Conference paper	✓	✓
	Benz et al. [6]	IEEE	Journal article	✓	✓
	Zhao et al. [88]	IEEE	Journal article	–	✓
	Babaei [5]	IEEE	Conference paper	–	✓
	Zhang et al. [85]	IEEE	Conference paper	✓	✓
	Chen et al. [11]	IEEE	Journal article	–	✓
	Liu et al. [42]	IEEE	Journal article	–	✓
	Tseng and Huang [66]	IEEE	Conference paper	–	✓
	Guo et al. [20]	IEEE	Journal article	–	✓
	Qin et al. [55]	IEEE	Conference paper	–	✓
	Huang et al. [24]	IEEE	Journal article	–	✓
	Yining et al. [79]	IEEE	Journal article	–	✓
	Zhao et al. [87]	IEEE	Journal article	–	✓
	Liao et al. [40]	IEEE	Conference paper	–	✓
	Kim et al. [30]	IEEE	Conference paper	–	✓
	Qiao et al. [54]	IEEE	Journal article	–	✓
	Chi et al. [14]	IEEE	Journal article	–	✓
	Hudson et al. [26]	Elsevier	Journal article	–	✓
	Landsmeer et al. [31]	Elsevier	Journal article	–	✓
2025	Zhao, Ding and Song [86]	Springer	Journal article	✓	✓
	Sahu et al. [59]	Springer	Journal article	✓	✓
	Nan et al. [49]	IEEE	Journal article	–	✓
	Shao et al. [60]	IEEE	Journal article	–	✓
	Wang et al. [69]	IEEE	Journal article	–	✓
	Morafah, Chang and Lin [48]	ACM	Conference paper	–	–

4.1. MQ1: What computing, storage and communication technologies are used in distributed edge AI architectures?

In the domain of EdgeAI, a vast array of computing, storage, and communication technologies has been employed to optimise distributed systems architectures. These technologies are pivotal for enhancing performance, efficiency, and scalability in resource-constrained environments typical of edge computing. A summary of these approaches, categorised by technology and their primary optimisation benefits, is presented in table 4.

Several studies have focused on specialised hardware accelerators and computing components to boost computational efficiency. For instance, Nunez-Yanez and Hosseinabady [51] explore the use of field-programmable gate arrays (FPGAs) to implement high-performance hardware accelerators for neural networks, specifically targeting dense and sparse matrix multiplications. This approach leverages FPGAs' ability to accommodate arbitrary arithmetic and sub-byte precision, which is essential for deploying neural networks on edge devices with limited resources. Similarly, Landsmeer et al. [31] evaluate cutting-edge AI accelerators – including NVIDIA graphics processing units (GPUs), Graphcore IPUs, GroqChips, and Google Tensor Processing Units (TPUs) – for simulating biologically realistic brain models. The study details the architectures, memory hierarchies, and communication mechanisms of these platforms, discussing how they handle the intensive computational and memory demands of large-scale simulations. The use of specialised cores, such as Tensor Cores and Sparse Cores, for optimised computations and data exchange is highlighted, showcasing advancements in computing technologies within distributed systems. This trend reveals a critical shift in EdgeAI architectures: by adopting PIM and CiM strategies, storage components effectively become active computing units. This shift not only addresses the "memory wall" bottleneck but also directly

Table 4
Summary of computing, storage, and communication technologies in distributed EdgeAI architectures (MQ1).

Approach category	Key technologies	Main benefits and optimisations	Key references
Hardware accelerators	FPGAs, GPUs, TPUs, IPUs, NPUs	Enables high-performance inference, supports arbitrary arithmetic (sub-byte precision), and accelerates matrix multiplications.	Nunez-Yanez and Hosseinabady [51], Landsmeer et al. [31], Minh, Mai and Minh [47]
Processing-in-Memory	ReRAM, CiM, PRAM, FeFET	Addresses the "memory wall" bottleneck, performs in-place matrix-vector multiplication, and reduces data movement energy.	Ogbogu et al. [52], Qu et al. [57], Takeuchi [64]
Heterogeneous computing	CPU + NPU Co-scheduling	Balances general-purpose processing with specialized neural tasks to optimise power and accuracy.	Lin et al. [41]
Task offloading	Edge-Cloud, D2D, Satellite-Edge	Mitigates resource constraints on end-devices by shifting heavy DNN tasks to more capable edge/cloud nodes.	Yun, Choi and Kim [80], Yun et al. [81], Hudson et al. [26], Zhao, Ding and Song [86]
Orchestration platforms	Docker, kubernetes, π-Edge	Abstracts hardware complexity, manages containerized services, and enables scalable deployment.	Chan et al. [9], Psaromanolakis et al. [53]
Distributed learning	Federated learning, AirComp	Enables collaborative training without raw data sharing; optimizes bandwidth via over-the-air computation.	Wang et al. [71], Xu et al. [75], Liao et al. [40], Liu et al. [45]

optimises the energy costs of internal communication, demonstrating how hardware choices in one pillar (storage) dictate efficiency in the others (computing and communication).

Processing-in-memory (PIM) accelerators have been investigated to address the bottlenecks between memory and processing units. Ogbogu et al. [52] explore the use of resistive random access memory (ReRAM)-based PIM accelerators to efficiently train graph neural networks (GNNs) on large-scale graph datasets. The paper addresses the challenges posed by the high computational and storage demands of GNN training on resource-constrained devices, proposing a novel data-pruning approach to enable high-performance, reliable training. In a similar vein, Qu et al. [57] discuss ReRAM-based PIM architectures that perform in-place matrix-vector multiplication, thereby significantly enhancing the computing efficiency of deep neural network inference tasks on edge devices. Additionally, Takeuchi [64] examines computation-in-memory (CiM) systems utilising non-volatile memories such as parallel random access machine (PRAM), ReRAM, ferroelectric field-effect transistor (FeFET), and flash memories for AI applications, serving both as computing and storage components within edge devices, enhancing performance and energy efficiency in distributed architectures.

Heterogeneous computing resources have been leveraged to optimise performance in edge environments. Lin et al. [41] discuss the integration of general-purpose Central Processing Units (CPUs) and specialised neural processing units (NPUs) in edge nodes to perform DNN inference tasks. By optimising the scheduling of DNN layers between CPUs and NPUs, the study demonstrates improvements in inference performance, accuracy, and power consumption. Moreover, Minh, Mai and Minh [47] highlight the use of NVIDIA Jetson Nanodevices as powerful yet energy-efficient computing platforms for running deep learning models in real-time on embedded systems. The Jetson Nano's low power consumption and powerful GPU make it ideal for EdgeAI applications.

Offloading tasks from edge devices with limited computing power to more capable edge servers has addressed this issue. Yun et al. [81] and Yun, Choi and Kim [80] highlight the challenges posed by

IoT devices' limited computing capabilities and memory constraints, which demand offloading DNN inference tasks to powerful edge servers. They delve into communication mechanisms between IoT devices and edge servers, including wireless channels, data transmission rates, and protocols such as hybrid automatic repeat request with chase combining (HARQ-CC) to improve data transmission reliability. Similarly, Hudson et al. [26] discuss a three-tier distributed architecture comprising mobile end-user devices, edge clouds, and remote central cloud servers. Edge clouds provide computing and storage resources to process user requests and host AI-based services. At the same time, communication technologies include device-to-device (D2D) communication between edge clouds, enabling offloading of requests when local resources are insufficient.

Software frameworks and management platforms are crucial in orchestrating distributed edge resources. Chan et al. [9] discuss implementing a cluster-based edge computing system that integrates Docker, Kubernetes, Prometheus, Grafana, and Node Exporter. These tools establish a containerised environment and manage the deployment and orchestration of services on heterogeneous edge devices, illustrating the importance of software solutions in distributed architectures. Similarly, Psaromanolakis et al. [53] introduce the π-Edge Platform, a cloud-native edge management platform that leverages container orchestration and a microservices architecture to manage edge resources efficiently. The platform uses Platform as a Service (PaaS) and Function as a Service (FaaS) delivery models to deploy and manage services at the edge, simplifying operations and reducing management overhead. Zhao, Ding and Song [86] describe a three-layer *cloud-edge-device* stack that couples IoT devices (local computing and minimal onboard storage), LEO-satellite edge servers (moderate computing/storage on flexible payloads) and ground station clouds (large data center resources). Communication relies on Ka-band satellite links, OFDMA multiple access and wired backhaul from the gateways to the core network. These software orchestration layers act as the binding mechanism that abstracts the underlying complexity of the heterogeneous hardware (CPUs, NPUs, and PIMs) discussed previously, allowing the logical data path to be optimised independently of the physical constraints.

FL has emerged as a key paradigm for distributed EdgeAI, enabling multiple edge devices to collaboratively train a global model without sharing raw data. Wang et al. [71] explore FL combined with over-the-air computation and reconfigurable intelligent surfaces (RIS) to increase communication efficiency and mitigate bottlenecks in FL systems. Xu et al. [75] provide a comprehensive overview of a distributed system architecture using Jetson Xavier NX embedded devices as edge nodes for real-time video processing, and 2080 Ti GPUs in a central server for intensive model training and aggregation tasks. The FL framework processes and stores video data locally on edge devices, reducing dependency on centralised storage and minimising data transmission needs. Additionally, Liao et al. [40] discuss deploying a heterogeneous system composed of NVIDIA Jetson devices as edge workers and a deep learning GPU workstation as the parameter server. The study highlights the challenges posed by resource constraints in distributed architectures and how they are managed through efficient communication enabled by Wi-Fi routers.

Similarly, Zhang et al. [83] explore FL architectures that distribute computation across multiple clients (edge devices) and a central server, discussing the implementation of hyperdimensional computing (HDC) models on edge devices and highlighting their compact model size and lower computational cost. Shao et al. [60] model a classic two-tier FL architecture in which edge clients with limited resources perform local gradient computations and keep residual information in their limited on-device storage, while a central parameter server with ample computing power and memory aggregates the shared updates over conventional IP links. The article emphasises that "communication latency between the server and client accounts for more than 70% of collaborative training time" and that bandwidth differences between devices can reach two orders of magnitude, making the upload channel the system's bottleneck. Morafah, Morafah, Chang and Lin [48] explain that their FL configuration combines small IoT devices, smartphones, PCs and servers, each with its own computing power and memory. These devices communicate with a central server via common internet links, and the server maintains a public, unlabeled dataset for knowledge sharing.

Communication technologies are vital for efficient data transmission in distributed systems. Hlophe,

Awoyemi and Maharaj [21] highlight the future use of 6G communication technologies to achieve high data transmission rates and low latency, which are crucial for supporting latency-sensitive applications such as virtual reality and gaming. Liu et al. [45] explore the use of AirComp as a communication technology in distributed EdgeAI sensing systems. By leveraging the superposition property of wireless channels, the AirPooling framework reduces the communication bottleneck caused by uploading high-dimensional data from multiple sensors to a server. Guo et al. [19] discuss integrating cloud infrastructure with edge networks, specifically highlighting 5G edge networks and New Radio technologies. The deployment of private 5G edge networks tailored for secure, private services in industrial Internet of Things (IIoT) applications illustrates advancements in communication technologies within distributed systems. Similarly, Ji and Qin [28] discuss a semantic task offloading system for semantic networks, employing communication technologies such as orthogonal frequency division multiple access (OFDMA) to facilitate task transmission between user equipment and edge servers.

Blockchain technology has been integrated into edge computing architectures to enhance security and data integrity. Qiu et al. [56] discuss the integration of edge computing and blockchain technologies within distributed systems architectures for beyond 5G (B5G) networks. Edge computing brings resources closer to data sources, reducing latency and bandwidth usage. At the same time, blockchain ensures decentralised and immutable records of learning outcomes, facilitating secure and efficient communication between edge nodes. Zhao et al. [88] integrate blockchain with FL within distributed edge intelligence (EI) architectures, using smart contracts to orchestrate FL training and ensure fairness and security during model training. Additionally, Abdel-Basset, Moustafa and Hawash [1] present a privacy-preserving cyberattack detection framework, Fed-Trust, designed for IIoT environments. The framework integrates edge nodes, fog servers, cloud servers, and a blockchain network to create a distributed system architecture that enhances security and performance.

Integration of AI models and applications at the edge has been explored to optimise performance. Kim et al. [30] discuss integrating semantic communication structures with orthogonal frequency division multiplexing (OFDM) systems in distributed architectures. It explores using multi-access edge computing (MEC) servers in base stations and user equipment to enable semantic communication for EI applications such as image segmentation. The article details how semantic encoders and decoders are implemented in the user equipment and the MEC server, using technologies such as vision transformers (ViT) for efficient data processing. Gong et al. [18] present an integrated blockchain-assisted satellite-ground digital twin (SG-DT) network model, incorporating computing technologies such as Low Earth Orbit LEO satellites and high altitude platforms (HAPs) to provide global coverage and powerful computing capabilities.

Moreover, Liu et al. [42] describe a network architecture where an edge device and an edge server, both equipped with Dual-Function Radar and Communication systems, collaborate to perform AI inference tasks. The paper specifies communication technologies, such as digital modulation and fixed-frequency carriers for data transmission, and computing technologies, including AI models like support vector machines (SVMs) and multilayer perceptron neural networks, deployed on edge devices and servers. Nan et al. [49] present a two-layer edge intelligence architecture consisting of battery-powered mobile devices and a co-located MEC server. The paper characterises the computational capabilities of each layer using CPU frequency variables for device and server processors, models the storage/queuing limits at the edge node, and details the communication substrate as an FDMA uplink with explicit bandwidth and power spectral density parameters.

Edge computing architectures also benefit from the use of Digital Twins and semantic communication. Qiao et al. [54] introduce a digital twin-enabled IIoT framework where digital twins provide real-time virtual representations of physical IIoT devices, enabling accurate simulations and intelligent decision-making. Computing technologies are explored by adjusting the CPU frequency in IIoT devices to optimise local computations. Communication technologies are central to the framework, with wireless communication parameters such as bandwidth rate and transmission power being dynamically adjusted to improve FL efficiency and accuracy in a distributed environment with limited resources. Furthermore, Lu et al. [46] introduce the concept of Digital Twin Wireless Networks,

where IoT devices are mapped to their digital twins on edge servers equipped with Mobile Edge Computing capabilities. This configuration leverages advanced computing and storage technologies at the network edge, and 6G wireless communication technologies improve connectivity and reduce latency between users and edge servers. Huang et al. [23] present the semantic data sourcing (SEMDAS) framework, leveraging semantic communications to increase the efficiency of data sourcing for EI applications, highlighting advanced wireless techniques such as AirComp, multi-access schemes, radio resource management, and beamforming.

In industrial applications and the IIoT, several studies have focused on integrating computing, storage, and communication technologies tailored for industrial environments. Liu et al. [43] discuss the computing and communication technologies used in industrial wireless networks within a distributed systems architecture. It highlights the roles of machine-type devices (MTDs), industrial base stations, edge servers, and cloud servers in processing large amounts of heterogeneous, compute-intensive, and delay-sensitive data generated by distributed MTDs in smart manufacturing environments.

Likewise, Xia et al. [74] introduce an IIoT architecture that incorporates hierarchical Software-Defined Network controllers and Radio Access Networks based on the Mobile Edge Cloud, utilising servers and virtualisation techniques to create virtual machines for IIoT device processing tasks. Askarizadeh, Morsali and Nguyen [4] explore computing, storage, and communication technologies in distributed systems, specifically in the context of transfer learning based on multi-source instances with limited resources, balancing trade-offs between model accuracy and resource consumption.

Simulation and testbeds have been instrumental in evaluating distributed systems architectures. Wang et al. [68] introduce SimEdgeIntel, an open-source simulator that models interactions between cloud servers, base stations, proxy servers, and mobile devices in a distributed system. The article sheds light on the computing, storage, and communication technologies utilised in distributed EI architectures by enabling the simulation of components such as limited storage capacities in base stations and devices, communication delays, and caching mechanisms. Tuli, Casale and Jennings [67] describe a heterogeneous testbed comprising Raspberry Pi 4B nodes (edge devices) and virtual machines provisioned on the Microsoft Azure cloud platform. The article highlights how the SimTune framework leverages this distributed architecture to optimise resource allocation, manage workloads, and improve overall system performance in edge and cloud computing environments.

Several studies have focused on optimising communication protocols and resource allocation. Zhou et al. [89] examine the communication technologies involved in edge computing architectures, particularly the allocation of wireless resources in time-division multiple access (TDMA)- based systems. It explores how limited wireless resources, such as time and power, are allocated among multiple users and learning tasks to optimise learning performance at the edge. Chen et al. [10] detail the computing and communication technologies employed in an EI network, discussing bandwidth allocation and uplink power management mechanisms to facilitate efficient data transmission between edge devices and the central server. Zhao et al. [87] emphasise the heterogeneity of resources in distributed edge servers, noting that these servers have varying computing and communication capabilities. The authors model communication delays, including propagation and transmission delays, and incorporate factors such as physical distance, bandwidth, and channel conditions.

Storage optimisation techniques are critical in managing limited resources in edge environments. Chen et al. [11] delve into model compression techniques, such as parameter pruning and knowledge distillation, to reduce the complexity and size of ML models deployed on edge devices. The article addresses storage constraints and enhances the efficiency of FL processes. Furthermore, Zhang et al. [85] discuss model quantisation techniques to reduce the memory footprint of large language models (LLMs), enabling their deployment on high-end servers with limited resources.

Advancements in memory and storage technologies have also been explored. Zeng et al. [82] discuss using advanced smart sensors with computing capabilities, such as integrating FPGA and ASIC chips (including TPUs, NPUs, and Vision Processing Unit (VPU)), and AI software that enables on-device ML processing. The article highlights the limited memory capacity of sensors, leading to challenges such as catastrophic forgetting because they cannot store complete data streams. Those articles underscore the importance of efficient storage solutions in distributed architectures.

In much the same way, Ogbogu et al. [52] address storage considerations by discussing the limited write endurance of ReRAM cells in PIM accelerators and proposing strategies to mitigate this issue, which are crucial for maintaining performance and reliability in distributed computing systems. Additionally, Chen et al. [13] discuss using intelligent endpoints with edge computing capabilities to improve the robustness of network topologies, enabling intelligent endpoints to contribute to network optimisation.

In multimedia applications, Chi et al. [14] focus on devices with limited computing capabilities that generate, pre-process, and compress high-definition video data before transmitting it to edge servers via wireless links. The computing aspect involves the local processing power of devices used for data compression and pre-processing. Communication technologies are highlighted through the allocation of wireless resources, such as bandwidth and transmission power, and the optimisation of wireless link transmission rates. Similarly, Ji and Qin [28] utilise GPUs in both user equipment and edge servers to handle task processing, specifically bidirectional encoder representations from transformers (BERT) models for machine translation tasks, demonstrating the integration of computing and communication technologies in distributed systems.

Integration of cloud and edge computing has been explored to enhance system performance. Yining et al. [79] discuss a "Cloud-Edge-Terminal" system in EI-enabled 6G networks, highlighting a distributed architecture comprising cloud servers, edge servers, and artificial intelligence of things (AIoT) devices. Computing tasks are distributed among these layers: cloud servers handle large-scale AI model training, edge servers perform model compression and resizing, and AIoT devices execute the models for semantic tasks. The study emphasises the importance of selective model transmission and adaptive resource allocation due to the finite storage and computing resources on board AIoT devices. Additionally, Guo et al. [20] details a distributed system architecture that leverages edge computing, wireless communication technologies, and FL to deliver VR content in the metaverse. The architecture demonstrates how computing and communication technologies are integrated into a distributed system to enhance efficiency and scalability.

Similarly, Sahu et al. [59] present a three-layer architecture in which battery-powered smartphones interact with nearby edge servers via Wi-Fi. At the same time, more intensive computational tasks or long-term data are relayed to a cloud backend. The mobile device layer hosts lightweight sensing and rendering; the edge layer provides GPU/CPU acceleration and caching; and the cloud layer provides additional storage and a model update service for learning policies. Wang et al. [69] describe a three-tier cloud-edge-device architecture in which high-performance cloud servers perform computationally heavy deep reinforcement learning (DRL) training, edge servers deployed in 110 kV substations host the inference engines for real-time control, and field sensors (smart meters, switch monitors and SCADA units) transmit operational data via standard IP links to the cloud for model updates.

Some studies focus on specific components or mechanisms within distributed architectures. Benz et al. [6] provide an in-depth exploration of direct memory access engines (DMAEs) architectures, essential computing and communication technologies in modern distributed systems. The paper presents the intelligent DMA (iDMA) architecture, highlighting its modular and highly parametric design that accommodates various on-chip protocols. This architecture facilitates efficient data movement between processing elements and memory hierarchies in different systems, enhancing computing and communication resources in distributed architectures. Tseng and Huang [66] explore using Bfloat16, a half-precision floating point format, as a computing technology to enhance FL systems. By converting model parameters to Bfloat16, the proposed FedBF16-Dynamic engine reduces computational resources and power consumption required for deep learning operations on edge devices.

Other studies address the challenges of resource heterogeneity and optimisation in distributed systems. Zhao et al. [87] emphasise the heterogeneity of resources in distributed edge servers, noting that these servers have varying computing and communication capabilities. The authors model communication delays, including propagation and transmission delays, and incorporate factors such as physical distance, bandwidth, and channel conditions. Li, Bi and Wang [34] explore an EI system

that integrates mobile edge computing and artificial intelligence, describing a distributed architecture in which end users collaborate with an edge server to train and update AI models using FL techniques. Communication resources, such as bandwidth and transmission power, are allocated to optimise transmission efficiency between users and the edge server.

In the same way, Sahu et al. [59] describe a five-layer stack in which Snapdragon-class devices provide lightweight sensing and rendering, high-end servers with 32-core Xeon CPUs cache data and perform GPU/CPU-heavy tasks, and a cloud backend provides large-scale storage as well as model update services; all layers are linked via 100 Mbps Wi-Fi and conventional Internet backhaul, thus detailing the heterogeneous computing, memory and wireless substrates that make up the distributed system.

Some articles provide isolated insights into specific technologies or frameworks. Desnos et al. [15] discusses the use of various computing technologies for ML inference in the context of edge and embedded systems, evaluating the performance of tangled program graphs (TPGs) across different hardware platforms, including embedded systems such as Raspberry Pi and Jetson Nano devices. Although the main focus is on accelerating inference through code generation, the article highlights the importance of selecting appropriate computing platforms to meet the constraints of embedded systems on distributed architectures.

Exploring computing, storage, and communication technologies in distributed systems architectures within the EdgeAI domain reveals diverse approaches and innovations. Figure 4 shows some of the leading technologies used in distributed systems, grouped into computing, storage, and communication. From specialised hardware accelerators and optimised software frameworks to advanced communication protocols and security enhancements, these studies collectively advance efficient and robust distributed EdgeAI systems. The integration of these technologies addresses the challenges posed by resource constraints, heterogeneous environments, and the need for low-latency, high-throughput data processing in edge computing scenarios.

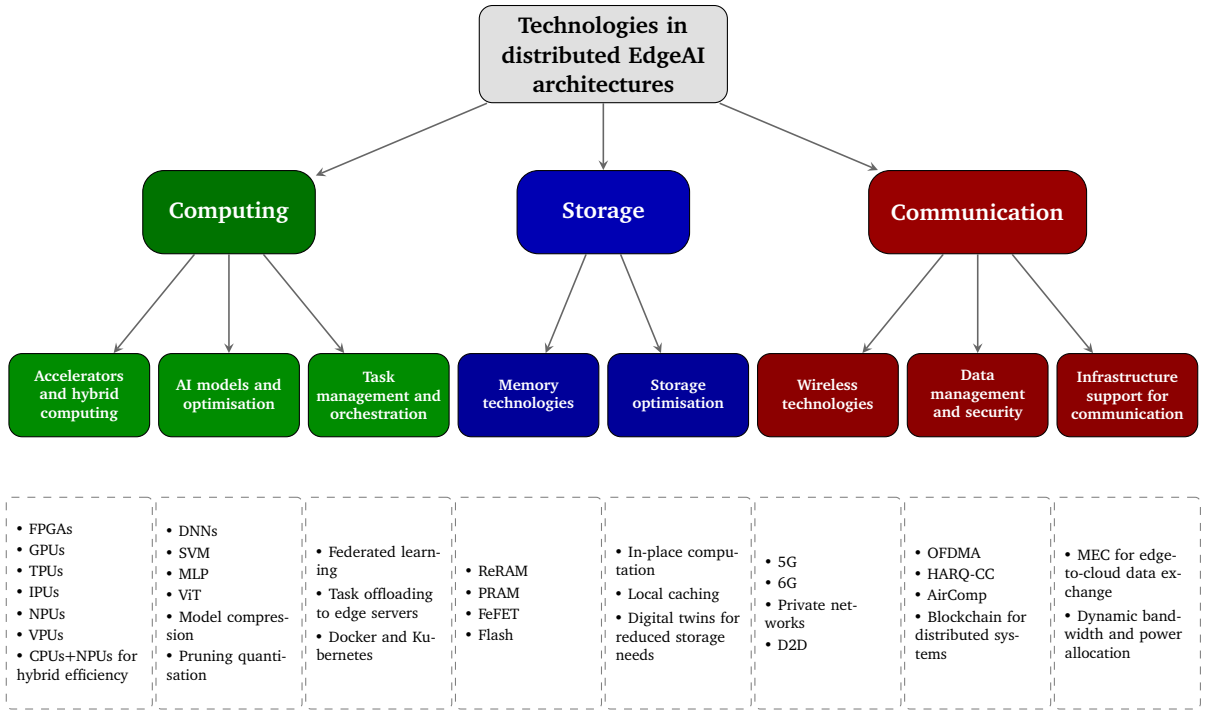

Figure 4: Key technologies enabling efficient and scalable distributed EdgeAI architectures.

4.2. MQ2: What optimisations can be made to the data path for AI workloads between edge-fog-cloud?

Efficient optimisation of the data path between edge, fog, and cloud layers is essential for enhancing the performance, efficiency, and scalability of EdgeAI systems. Optimising in this context refers to improving and fine-tuning how data is transmitted, processed, and managed across these layers. Optimisation involves implementing strategies that reduce communication overhead, minimise latency, and efficiently use bandwidth and energy resources. By carefully optimising data flow, systems can handle large volumes of data more effectively and respond quickly to real-time demands. Various strategies have been proposed in the literature to address challenges such as communication overhead, latency, bandwidth limitations, and energy consumption.

Several studies focus on optimising data transmission in FL setups to improve the data path between edge devices and servers. For instance, Wen et al. [72] present the RFA-RFD framework. This two-step FL approach generates independent and identically distributed datasets locally using conditional variational autoencoders. It employs knowledge distillation to minimise the amount of data transmitted, reduce communication overhead, and enhance data privacy. Similarly, Xu et al. [75] introduce the anonymous and efficient federated learning (AEFL) strategy, which standardises and minimises model parameter sizes and selectively retrains specific model branches, thereby significantly reducing data exchange and bandwidth usage. To address communication bottlenecks, Lian et al. [38] propose AGQFL, dynamically adjusting the quantisation precision of model gradients transmitted from edge devices to the cloud, effectively reducing communication overhead and mitigating bandwidth heterogeneity. Additionally, Tseng and Huang [66] introduce FedBF16-Dynamic, optimising the data path by converting model parameters to Bfloat16 format and implementing dynamic upload schemes that adapt to network bandwidth heterogeneity across edge devices. Furthermore, Zhang et al. [83] discuss optimisations in the communication pathway by comparing hypervector aggregation and associative memory aggregation in HDC-based FL; the associative memory aggregation method significantly reduces communication overhead by transmitting smaller model updates, streamlining the data path between edge devices and the central server. Shao et al. [60] presents FedLoRE. This model replaces the total gradient exchange with a low-rank difference of each client's gradient, while retaining the local residual components. This design generates savings of up to 89% in communication costs and reduces total training time by more than 80% compared to state-of-the-art customised FL baselines. Morafah, Chang and Lin [48] propose FedHD, which skips sending complete model weights. Instead, each device sends only the "logits" (the model's output scores) to the server; the server then distils them into a global model for each group using a clever weighting trick that gives more decision-making power to the larger models.

Optimising data transmission through edge caching strategies and communication models is explored in Wang et al. [68], which proposes caching popular content at edge nodes, such as base stations and mobile devices, to reduce backhaul traffic and content delivery time. The paper discusses a learning-based caching algorithm using federated deep reinforcement learning to enhance cache replacement strategies, improving the efficiency of data flow in edge-Fog-Cloud architectures by minimising unnecessary data transmissions and leveraging local computations.

To address the challenges of high-dimensional model transmissions on limited radio resources, Wang et al. [71] present AirComp as a solution for spectrum-efficient uplink model aggregation. Integrating RIS improves signal alignment and reduces aggregation errors caused by fading and channel noise. Joint optimisation of the AirComp transceiver and RIS phase shifters is crucial for minimising transmission distortions and improving overall FL performance. Similarly, Liu et al. [45] introduce the AirPooling framework, enhancing data transmission efficiency from end sensors to the server by reducing latency and bandwidth requirements through AirComp, simplifying the data path in distributed sensing systems.

Architectural optimisations play a significant role in enhancing the data path. Benz et al. [6] present a modular and highly parametric Direct Memory Access Engine architecture designed to optimise data movement across edge, fog, and cloud layers. Strategies such as minimising hardware buffering,

maximising bus utilisation, and implementing multi-stage transfer acceleration schemes are explored. Transfer descriptor chaining and in-stream acceleration reduce latency, improve throughput, and lower energy consumption. Resource allocation and scheduling are addressed by Li et al. [35], which introduces a joint data partitioning and rate control design to improve learning accuracy and energy efficiency by minimising communication load and optimising resource allocation for multiple learning tasks. Liao et al. [40] propose the MergeSFL framework, introducing resource fusion and batch-size regulation to improve data transmission efficiency between edge devices and a parameter server, thereby reducing communication overhead and optimising the data path. Sahu et al. [59] propose a Deep-Q-Network policy that jointly chooses where each AR task is executed (local, edge, cloud) and how much bandwidth/CPU to allocate, while an adaptive quality module reduces texture sampling when resources become scarcer; together, these mechanisms cut redundant uploads and reduce round-trip traffic, simplifying the entire device-edge-cloud path. The structure proposed by Wang et al. [69], divides optimisation into centralised DRL training in the cloud and decentralised inference at the edge; in addition, it introduces a new distribution network partitioning algorithm and an exchange importance metric that jointly reduce the state/action spaces seen by each edge server, thus reducing the volume and frequency of parameter exchanges between layers.

Optimising collaboration between devices and edge computing components is crucial for improving the data path. Li et al. [33] present the Edgent framework, which mitigates latency and power consumption issues inherent in cloud-centric approaches by adaptively managing the distribution of computational tasks based on available bandwidth through DNN partitioning and right-sizing. In a similar vein, Chi et al. [14] propose a source-value-based resource allocation scheme that jointly optimises task data generation rates, compression rates, bandwidth allocation, and transmission power, using the Timeliness-Accuracy Degradation metric to ensure only the most valuable and timely information is transmitted, reducing unnecessary network congestion.

System models that facilitate efficient data flow across layers are essential. Xu, Qian and Hu [76] explore data path optimisations by illustrating how edge computing nodes act as intermediaries, handling data aggregation and preprocessing before sending relevant information to the cloud, reducing communication overhead and latency. Yining et al. [79] introduce an intelligible model transmission framework for the Cloud-Edge-Terminal ecosystem, emphasising model compression at edge servers to generate scalable AI models suitable for AIoT devices with varying resource constraints, thereby reducing transmission latency and overhead between cloud servers, edge servers, and terminals.

Specific applications also drive data path optimisations. Guo et al. [20] focus on virtual reality systems and implement a multi-view synthesis model using an FL framework to reduce the volume of data transmitted from the base station to users. By transmitting a sparse set of input views and allowing users to locally synthesise their specific viewports, bandwidth consumption and transmission latency are decreased. Optimising data transmission using intelligent reflecting surfaces is presented in Huang et al. [24]. By activating the signal-propagation environment, an intelligent reflecting surface (IRS) mitigates mutual interference between radar sensing and image data download, thereby enhancing both radar sensing quality and image analysis accuracy. Optimising IRS phase shifts, along with image resolution and transmission power, results in a more efficient data path in the edge computing environment.

Optimisations in computation offloading processes are explored by Gong et al. [18], who introduce a multi-agent deep federated reinforcement learning algorithm to optimise task scheduling, resource allocation, and energy harvesting across IoT devices, airborne base stations, and satellite servers. Managing dynamic channel gains, stochastic task arrivals, and volatile air base station locations improves data flow and computation offloading decisions in the edge-fog-cloud continuum. Abdel-Basset, Moustafa and Hawash [1] address security and efficiency in data path optimisations and propose a blockchain-orchestrated EI system within the Fed-Trust framework. The framework reduces latency by offloading computational workloads, such as mining and verification, from edge nodes to more capable fog servers. It improves the efficiency of data transmission across the edge, fog, and cloud layers, enhancing communication flow and resource utilisation. The contribution of

Zhao, Ding and Song [86] is a joint optimisation of cross-region task offloading and cross-domain resource allocation, spanning devices, satellite edges, and terrestrial clouds. Formulated as an MDP, the proposed spatio-temporal attention PPO policy (STA-PPO) decides, per task, where to execute and how much bandwidth, power and CPU to allocate, so that end-to-end delay is minimised and throughput is maximised. Nan et al. [49] proposed a robust joint optimisation framework that determines which tasks to offload and how much uplink bandwidth and MEC CPU frequency each offloaded task receives, thereby minimising the worst-case energy-time cost metric. Closed-form solutions for local execution and a geometric programming/SCA solver for edge execution are combined with a coordinate descent search over the binary offload vector.

In summary, optimising the data path between edge, fog, and cloud layers is paramount for enhancing the efficiency, scalability, and responsiveness of EdgeAI systems. The strategies explored encompass a range of techniques aimed at reducing communication overhead and latency while efficiently utilising bandwidth and energy resources. One significant approach is to enhance FL frameworks to minimise data transmission. This is achieved by generating local datasets that align with global models, compressing model parameters, and selectively updating specific model branches. Such methods reduce the volume of data exchanged between edge devices and servers, bolster data privacy, and adapt to varying network conditions.

Another set of innovations focuses on AirComp and advanced signal processing to improve data aggregation and transmission efficiency. By leveraging techniques like RIS and intelligent reflecting surfaces, these solutions enhance signal alignment, mitigate interference, and reduce aggregation errors caused by channel noise. Architectural optimisations, including modular engine designs and efficient resource allocation schemes, are crucial in optimising data movement and processing across the network layers. These designs implement strategies like minimising hardware buffering, maximising bus utilisation, and dynamically adjusting resource allocation to meet the demands of multiple learning tasks. Application-specific optimisations further tailor the data path enhancements to specific use cases, such as virtual reality systems and distributed sensing, by enabling local data synthesis and intelligently configuring signal-propagation environments. Collectively, these multifaceted strategies address the critical challenges of communication overhead, latency, bandwidth limitations, and energy consumption, significantly improving the performance and scalability of EdgeAI systems in real-world scenarios.

Critically, this review highlights that data path optimisation is rarely achieved by improving transmission protocols in isolation. Instead, it demands a dynamic trade-off: increasing local **computing** (via compression or Federated Learning) drastically reduces **communication** loads, while enhanced **storage** (through edge caching) eliminates redundant transmissions. Consequently, the most effective 'data path' is not necessarily the fastest link, but the one that best balances these three pillars to minimise total latency and energy expenditure. Figure 5 illustrates the Edge-Fog-Cloud architecture not merely as a hierarchy of hardware but as a stratified optimisation pipeline. At the **Edge layer**, techniques such as quantisation and Bfloat16 are critical to fit inference tasks within strict power envelopes, enabling local decision-making. Moving up, the **Fog layer** acts as a semantic filter, utilising Edge Servers and Federated Learning aggregators to process data locally, thereby preventing uplink bandwidth saturation. Finally, the **Cloud layer** is reserved for computationally intensive tasks such as centralised model training. The communication links between layers form an optimised data path, with protocols such as AirComp and 5G dynamically selected to balance the latency-accuracy trade-off.

4.3. SQ1: What ML optimisations can be made to improve network flow in an EdgeAI system?

Several studies have explored using FL and model compression techniques to reduce communication overhead and improve network flow in EdgeAI systems. For instance, Zhao et al. [88] introduced a framework that leverages a pre-trained corruption-detection model in an FL setup, improving data integrity checking while minimising communication between data owners and edge nodes. Similarly, Wen et al. [72] and Xu et al. [75] implemented data augmentation using conditional

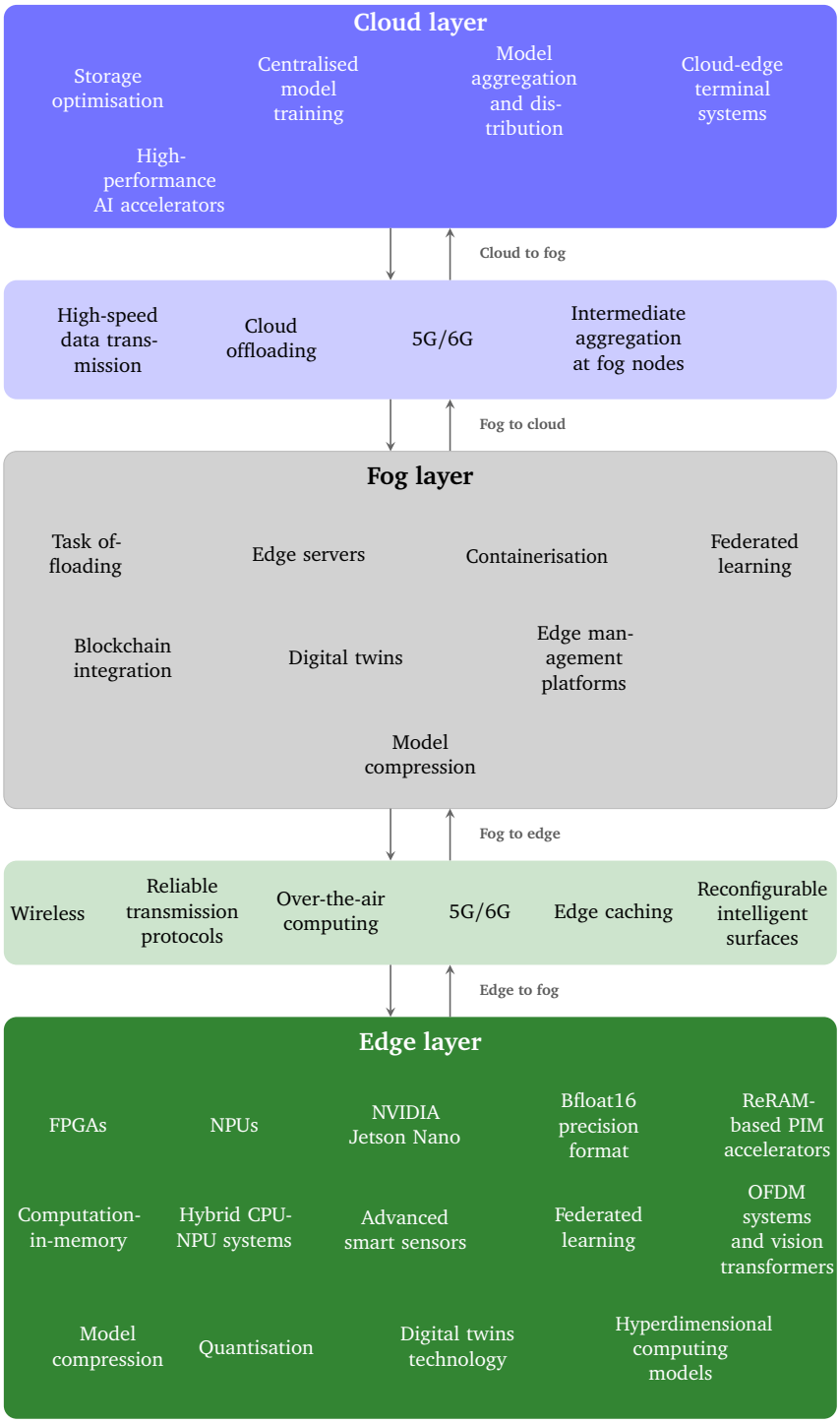

Figure 5: Optimisations techniques in edge-fog-cloud layers.

variational autoencoders (CVAE) and knowledge distillation to generate synthetic IID datasets locally, thus reducing the volume of data transmitted and improving network throughput. To further reduce communication overhead, Lian et al. [38] proposed gradient quantisation and adaptive precision adjustment within the AGQFL framework, compressing gradients and dynamically adjusting quantisation based on model convergence and bandwidth constraints. Random pruning techniques were employed by Chen et al. [10] and Chen et al. [11] within FL frameworks to reduce model size, thereby decreasing data transmission between edge devices and servers. The LotteryFL framework presented by Li et al. [32] leverages the Lottery Ticket hypothesis to identify sparse subnetworks within

larger neural networks, transmitting only essential parameters during federated updates. Additionally, Yining et al. [79] discussed model compression methods like quantisation, pruning, and knowledge distillation, along with flexible model selection strategies to optimise the transmission of intelligent models. Using federated multi-view synthesis models by Guo et al. [20] allows users to generate necessary VR content locally, reducing the need for extensive data transmission. Furthermore, Zhang et al. [83] introduced HDC-based FL with associative memory aggregation, reducing the transmitted data by aggregating model parameters rather than raw encoded hypervectors. Along the same lines, Babaei [5] detailed methods such as quantisation, weight pruning, and weight clustering to compress neural network models, facilitating faster inference and lower power consumption on edge devices and indirectly improving network throughput by minimising data transmission. The FedLoRE model by Shao et al. [60] is a machine learning optimisation: an online alternating minimisation algorithm learns low-rank subspaces of successive gradients by exploiting their similarity between rounds. By loading only these compact subspaces, the framework speeds up convergence, resulting in up to 94% less training time on some datasets and increasing network throughput, as far fewer parameters cross the link each round. Morafah, Chang and Lin [48] proposed FedHD, a tweak of ML: it uses knowledge distillation and adaptive weights to allow very different devices to teach themselves without large uploads. Experiments with CIFAR-10/100 show greater accuracy with far fewer bytes moved, proving that the ML trick really does increase the network's effective throughput.

In cooperative DNN inference, Yun, Choi and Kim [80] and Yun et al. [81] introduced methods for partitioning the inference process between IoT devices and edge servers. They utilised knowledge distillation to create smaller, memory-efficient student DNNs from larger teacher models, optimising data flow by reducing computational and memory loads on edge devices. Similarly, Li et al. [33] applied DNN partitioning and right-sizing to the Edgent framework, which adjusts the inference process based on real-time network conditions to improve network flow by balancing computation and communication. Liu et al. [45] modified the MVCNN architecture to incorporate AirPooling, enabling pooling of resources from multiple sensors directly over the air, reducing communication latency and bandwidth usage.

Reinforcement learning (RL) has been applied to optimise resource management and task offloading in EdgeAI systems, thereby improving network flow. Hlophe, Awoyemi and Maharaj [21] implemented deep Q-learning network (DQN) strategies, leveraging DRL to dynamically allocate tasks between local processing and MEC servers, accounting for real-time network conditions such as interference and congestion. Ji and Qin [28] employed a multi-agent proximal policy optimisation (MAPPO) algorithm to make intelligent decisions on task offloading, transmission power, and computation frequency, reducing communication latency and energy consumption. Lu et al. [46] proposed a multi-agent reinforcement learning algorithm to optimise edge association and bandwidth allocation, improving network efficiency and reducing latency. Furthermore, Gong et al. [18] utilised deep federated reinforcement learning (DFRL) within the LST-MADFRL algorithm to optimise computation offloading decisions, CPU cycle frequency, and energy-harvesting strategies, thereby improving network throughput and performance. Qiao et al. [54] introduced an Adaptive FL algorithm for IIoT-enabled Digital Twins, leveraging Deep Reinforcement Learning to dynamically adjust wireless parameters, reducing communication delays and power consumption. Similarly, Chi et al. [14] presented a knowledge-assisted dimension-refined reinforcement learning (DRRL) algorithm to optimise resource allocation, improving network efficiency and flow. Still in the field of ML, Zhou et al. [89] proposed the Learning centric wireless resource allocation (LCWRA) scheme, which allocates transmission time and energy to maximise learning performance across multiple tasks. Sahu et al. [59] present an RL mechanism that observes battery, channel rate and task complexity and then learns offloading actions that minimise long-term energy latency cost; experiments show energy savings of approximately 30% and a delay of <80 ms, confirming that the ML policy directly improves network throughput in an EdgeAI configuration.

Dynamic scheduling, resource allocation, predictive modelling, and proactive adaptation are crucial for optimising network flow in EdgeAI systems. Guo et al. [19] proposed the DISCO algorithm, which maximises the average size of scheduled data per communication round by prioritising devices based

on data availability, communication capabilities, and computational resources, thereby reducing transmission delays. Similarly, Liao et al. [40] introduced techniques such as resource merging, batch-size regulation, and genetic algorithms for worker selection to reduce the volume of transmitted data and optimise participation, alleviating communication bottlenecks and improving network throughput. Furthermore, Psaromanolakis et al. [53] employed MLOps services within the π-Edge Platform, using time-series forecasting and ML models such as N-BEATS, long short-term memory (LSTM), and support vector regression (SVR) to predict resource utilisation metrics. This enables proactive sizing and adaptation of services, ensuring efficient resource allocation and enhanced network flow. Likewise, Tuli, Casale and Jennings [67] presented SimTune. This framework uses neural network-based surrogate models to predict QoS metrics and inform scheduling decisions, optimising task allocation and improving network throughput.

ML optimisations have been applied to enhance network security and anomaly detection in EdgeAI systems, thereby indirectly improving network performance by ensuring reliable communication. Li et al. [37] utilised LSTM and SVM models to detect Man-in-the-Middle (MITM) attacks in Communications-Based Train Control systems, thereby enhancing the system's ability to recognise malicious activity in real time and optimise communication resources. Xu, Qian and Hu [76] presented supervised and semi-supervised learning models, including decision trees, bagging, one-class classification, and ensemble learning, to improve network flow analysis and anomaly detection. Abdel-Basset, Moustafa and Hawash [1] implemented a semi-supervised temporal convolutional generative adversarial network (TCGAN) within the Fed-Trust framework for cyberattack detection, applying group normalisation layers to reduce gradient diversity during training, thereby speeding up convergence and improving network throughput. Chen et al. [13] proposed an asynchronous distributed learning strategy using a deep reinforcement learning algorithm to optimise network topology, dynamically adapting node connections based on network state, improving network robustness and flow.

Advanced ML models, such as GNNs, have optimised network configurations in EdgeAI systems. Wang et al. [71] developed a GNN-based learning algorithm to map channel state information (CSI) directly to optimal AirComp transceiver configurations and RIS phase shifts, leveraging GNNs' ability to handle interactions between multiple edge devices efficiently, thereby improving signal alignment and noise suppression. Kim et al. [30] proposed a reinforcement learning-based pilot allocation algorithm and a masked autoencoder (MAE)-based channel estimator to enhance channel estimation and pilot signal allocation in semantic communication frameworks, enabling more accurate restoration of distorted information transmitted over wireless channels and thereby optimising network flow.

Hierarchical and decentralised learning frameworks have been proposed to reduce communication overhead in EdgeAI systems. Yang et al. [77] introduced E-Tree Learning, a decentralised framework that organises edge devices into a hierarchical tree structure for localised model aggregation, reducing overall communication compared to centralised approaches. Xia et al. [74] proposed partitioning neural networks for hierarchical deployment across the cloud, edge servers, and IIoT devices, with feature extraction using deep learning at the network edge to reduce data size and minimise communication overhead. Liu et al. [43] presented a MADRL-based resource allocation algorithm that enables distributed agents to cooperate and make decisions based on local observations without requiring complete system information, thereby optimising resource allocation and improving network flow.

Transfer learning and multisource optimisation have reduced resource consumption in EdgeAI systems. Qiu et al. [56] introduced the AI-Chain framework, leveraging FL and transfer learning to optimise the training and parameter sharing across edge nodes, reducing redundant computations and improving network throughput. Askarizadeh, Morsali and Nguyen [4] presented the multisource resource-constrained optimised transfer learning (MSOPTL) model, which formulates transfer learning as an optimisation problem balancing model accuracy and resource consumption, incorporating resource constraints directly into the learning process to reduce unnecessary data transmission and computation, thereby improving network flow.

Semantic matching techniques have minimised unnecessary data transmission and improved network flow. Huang et al. [23] introduced a framework using AI algorithms, such as semantic em-

beddings, uncertainty assessment, distribution matching, and autoencoders, to identify semantically relevant data sources, ensuring only relevant data is transmitted and processed. Similarly, Zeng et al. [82] presented the HFedMS system, incorporating the Layer-wise Alternative Synchronisation Protocol and a Semantic Compression and Compensation mechanism, which reduces communication overhead by synchronising important parameters more frequently and transmitting compressed historical data semantics.

Synthesising these findings, we observe a fundamental paradigm shift: network flow optimisation in EdgeAI is transitioning from purely protocol-based improvements to computation-driven reductions. By employing ML techniques such as semantic compression, deep reinforcement learning, or federated distillation, the system effectively trades "computing cycles" for "bandwidth savings". This confirms that, in modern EdgeAI, the most effective network optimiser is often a more sophisticated computational model, further blurring the distinction between the Communication and Computing pillars.

Figure 6 summarises the main optimisations highlighted by the studies, demonstrating a wide range of ML techniques to improve network flow in EdgeAI systems. These optimisations include FL enhancements, model compression, data augmentation, reinforcement learning for resource management, and advanced ML models for network configuration. These approaches contribute significantly to more efficient and scalable EdgeAI deployments by addressing issues such as communication overhead, computational efficiency, and resource allocation.

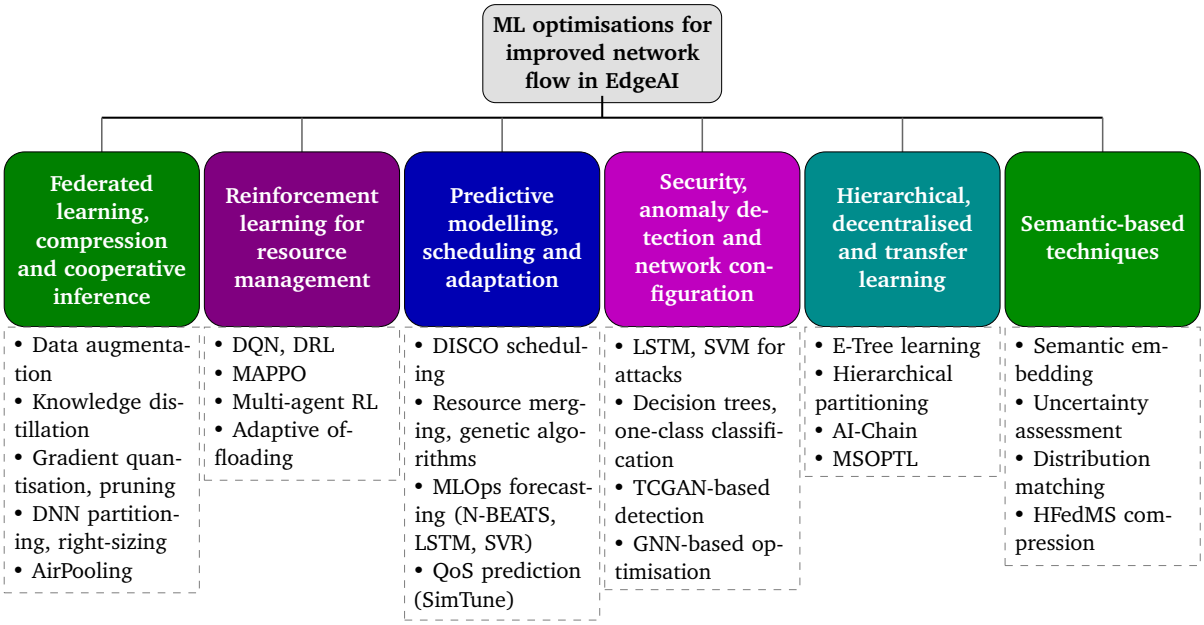

Figure 6: ML optimisation summary for improving network flow in EdgeAI systems.

4.4. SQ2: How can energy profiling be used to identify and implement energy-efficiency optimisations in EdgeAI systems, accounting for communication, storage, and computing limitations?

Addressing SQ2, various studies have explored the role of energy profiling in identifying and implementing energy-efficiency optimisations in EdgeAI systems, particularly under constraints on communication, storage, and computing resources. These efforts span hardware and architectural optimisations, FL and resource-allocation strategies, task offloading mechanisms, and energy-harvesting techniques.

Several researchers focus on enhancing energy efficiency through hardware-level innovations and architectural design choices. For instance, Landsmeer et al. [31] measure and compare the power

consumption of various AI accelerators during brain model simulations, providing foundational insights into energy consumption at the hardware level. Qu et al. [57] and Wu et al. [73] present strategies that significantly reduce energy consumption in specialised hardware for EdgeAI. Qu et al. [57] introduce a coordinated model pruning and mapping framework for ReRAM-based DNN accelerators, achieving substantial energy savings and reductions in chip area by optimising neural network weight mapping and introducing structured bit-level pruning. Similarly, Wu et al. [73] propose compressing sparse data and adopting compute-in-memory architectures with a zero-skipping circuit to reduce memory footprint and external data transactions, effectively minimising energy consumption associated with data movement. Additionally, Benz et al. [6] discuss how the modular DMAE architecture, specifically iDMA, contributes to power consumption optimisations by minimising hardware buffering and maximising bus utilisation. The paper highlights design choices that reduce static power consumption and improve efficient data movement, both crucial for power-constrained EdgeAI environments. Through case studies, it demonstrates how energy-efficient DMAE configurations can substantially reduce area and power overheads while maintaining high performance.

Approximate computing is another avenue explored to improve energy efficiency. Takeuchi [64] leverages the error tolerance of ML algorithms to tolerate some degree of memory-cell errors, optimising performance, power consumption, and cost in CiM systems. This co-design approach optimises performance and energy consumption while accounting for the storage and computational limitations inherent in EdgeAI devices. Additionally, Lin et al. [41] propose an online optimisation framework that incorporates energy profiling data to guide the scheduling of DNN layers between the CPU and NPU on edge devices, aiming to minimise inference accuracy loss while adhering to performance constraints and a power cap.

Optimising energy consumption through FL and intelligent resource allocation is another significant area of research. Wang et al. [68] introduce a federated deep reinforcement learning scheme to optimise content caching in edge-computing-supported IoT architectures, reducing communication overhead and latency by enabling edge devices to collaboratively train models without transmitting large datasets. Similarly, Li et al. [35] and Zhou et al. [89] integrate energy consumption as a key metric alongside learning accuracy within EdgeAI systems. Li et al. [35] address communication limitations by defining data transmission constraints and accounting for storage limitations via a limited data buffer capacity. It calculates and optimises energy consumption based on critical factors by incorporating these constraints into the optimisation process and performing energy profiling. Zhou et al. [89] formulate an optimisation problem that includes energy consumption limits for IoT devices, optimising transmission power and time allocation to maximise learning performance while adhering to energy budgets.

Dynamic scheduling policies are also employed to balance energy consumption with learning performance. Guo et al. [19] formulate an energy model that accounts for local update energy and uplink transmission energy for each device. The proposed algorithm ensures that devices do not exceed their energy budgets while minimising global loss by introducing a virtual energy queue and leveraging the Lyapunov optimisation framework. Furthermore, Chen et al. [10] and Chen et al. [11] formulate optimisation problems that jointly determine parameters such as pruning rate, CPU frequency, uplink power, and bandwidth allocation for selected edge devices, minimising total energy consumption during the FL process.

Task offloading mechanisms that incorporate energy profiling are crucial for optimising energy consumption. Hlophe, Awoyemi and Maharaj [21] integrate energy profiling into the task offloading mechanism, quantifying energy consumption for local computation and offloading. The proposed framework prioritises energy-efficient task allocation by evaluating the energy costs of processing tasks locally versus offloading them to MEC servers, thereby reducing energy consumption compared to traditional methods. Similarly, Ji and Qin [28] address energy consumption optimisation by modelling energy consumption for local and remote task processing and propose an algorithm based on multi-agent reinforcement learning to manage resources and minimise total energy consumption. Additionally, Liu et al. [42] focus on minimising energy consumption in EdgeAI systems through

an integrated sensing, communication, and computation framework. It introduces a task-oriented mode-selection scheme that dynamically selects between cooperative inference modes at the device, the server, and the edge device based on the energy profile. By evaluating energy costs across different operating modes under varying accuracy and latency constraints, the framework effectively identifies the most energy-efficient configuration for specific tasks, accounting for limitations in communication bandwidth, computing capabilities, and storage resources.

Energy profiling is also leveraged in optimisation algorithms to enhance energy efficiency. Binucci et al. [7] present an energy model that integrates both transmission and computational energy consumption, formulating an optimisation problem to minimise overall energy consumption while respecting latency and accuracy constraints. Employing stochastic optimisation tools demonstrates how the energy profile can inform dynamic resource allocation policies. Likewise, Li, Bi and Wang [34] incorporate the energy profile into the optimisation problem by quantifying the energy consumption of various processes and developing a method to identify optimal resource allocation strategies that minimise overall energy consumption. Furthermore, Qiao et al. [54] incorporate the edge server's energy profile in optimising resource consumption in EdgeAI systems. Recognising that FL processes consume significant energy, it proposes the Adaptive Federated Deep Transfer learning algorithm. This algorithm uses energy consumption data on CPU usage and transmission power to inform dynamic adjustments to wireless parameters. By formulating an optimisation problem that minimises the FL model's loss under energy and other resource constraints, the algorithm effectively identifies opportunities to reduce energy consumption while accounting for the communication, storage, and computing limitations inherent in resource-constrained IIoT devices.

Efficient energy management through harvesting techniques is another area of focus. Gong et al. [18] address the limited battery capacity of aerial base stations by proposing solar energy harvesting to supplement power needs. The optimisation problem includes constraints on power consumption and battery levels, balancing computing power consumption with energy harvesting to ensure power constraints are met over time.

Some studies specifically address energy consumption optimisations in the context of communication and storage constraints. Ogbogu et al. [52] present a data-pruning-enabled GNN training approach that reduces the number of subgraphs and write operations in ReRAM-based PIM accelerators, thereby decreasing the computational workload and energy consumption during training. Similarly, Zhang et al. [84] incorporate energy consumption considerations into optimising resource allocation and beamforming in EdgeAI systems, introducing a general energy consumption constraint into the problem formulation to minimise energy usage while meeting performance requirements. Additionally, Huang et al. [24] formulate a joint optimisation problem that includes constraints on image sensor energy consumption, modelling energy used for image preprocessing and transmission offloading. By incorporating energy constraints into the optimisation framework, it identifies optimal settings for IRS phase shifts, image resolution, and transmission power guided by the energy profile.

Some studies focus on optimisation strategies under strict power constraints. Wang et al. [70] propose a learning-centric power allocation strategy under a total power budget constraint to minimise classification error in EI systems, optimising transmission power between users to improve learning performance efficiently. However, it does not consider power consumption optimisations related to storage and computation limitations. In contrast, Tuli, Casale and Jennings [67] collect energy consumption data from edge devices and cloud servers to inform a surrogate model predicting QoS metrics, including energy usage. Incorporating energy profiles from various hardware configurations and workloads enables informed scheduling decisions that minimise energy consumption while accounting for communication and computational constraints.

Additionally, Yang and Chen [78] propose the spike-based nonlinear information bottleneck framework to increase the energy efficiency of spiking neural networks in EdgeAI systems. Introducing strategies that optimise the balance between information retention and compression improves robustness against noise and significantly reduces energy consumption compared to conventional methods.

These collective studies demonstrate that energy profiling is pivotal for identifying and implement-

ing optimisation strategies in EdgeAI systems. By addressing inherent limitations in communication, storage, and computing, researchers have developed diverse solutions, ranging from hardware innovations and resource-allocation algorithms to task offloading mechanisms and energy-harvesting techniques. These approaches contribute significantly to energy-efficient deployments by leveraging energy profiling not merely as a monitoring tool but as a unifying metric. By translating disparate resources such as bandwidth, CPU cycles, and memory access into a common energy cost, profiling enables the system to make objective trade-offs. This capacity to determine whether it is more efficient to cache a model locally or to request it from the cloud repeatedly ensures sustainability in resource-constrained environments.

4.5. SQ3: How can high-performance computing techniques be used to increase computing and communication efficiency in EdgeAI systems?

Several studies have leveraged specialised hardware accelerators, advanced architectures, compiler optimisations, and model compression techniques to enhance the efficiency of computing and communication in EdgeAI systems. For instance, Nunez-Yanez and Hosseinabady [51] present the development of specialised hardware accelerators for dense and sparse matrix multiplications using FPGAs. By investigating multi-precision arithmetic, quantisation, and pruning methods, the authors optimise neural network computations on edge devices, achieving significant performance improvements while managing resource constraints. This combination of hardware acceleration and model compression techniques exemplifies the synergy between specialised hardware and algorithmic optimisations.

Similarly, Landsmeer et al. [31] employ ML libraries such as TensorFlow alongside AI accelerators, including GPUs, IPUs, TPUs, and GroqChips. Through compiler optimisations using the XLA compiler for operation fusion and minimising data transfer overheads, the study accelerates simulations of complex brain models, illustrating how specialised hardware combined with optimised software can significantly improve computational efficiency. Proposing compiler optimisation techniques, Desnos et al. [15] demonstrates how code generation and compiler optimisations can significantly increase computing efficiency. Translating pre-trained Tangled Program Graphs into standalone C code eliminates overheads associated with dynamic structures and instruction decoding, resulting in inference times that are significantly faster than those of traditional methods. Moreover, Tseng and Huang [66] apply HPC techniques by adopting Bfloat16, a mid-precision floating-point format optimised for deep learning, reducing computational demands and power consumption during model training on edge devices.

Advancing this approach, Ogbogu et al. [52] and Qu et al. [57] utilise ReRAM-based PIM accelerators to enhance computing efficiency. Ogbogu et al. [52] propose a novel Binary Graph Classifier for subgraph pruning, leveraging the parallelism and in-memory computation capabilities of ReRAM-based PIM architectures to reduce computational workload and memory requirements during GNN training, thereby improving energy efficiency. Meanwhile, Qu et al. [57] integrate an automatic, structured bit-pruning method that employs reinforcement learning to determine optimal pruning strategies for neural network submatrices, effectively compressing neural networks and optimising their deployment on ReRAM-based accelerators.

Model compression and pruning techniques are widely adopted to enhance computational and communication efficiency. Yining et al. [79] explore methods such as quantisation, pruning, and knowledge distillation to reduce computational and storage demands on AIoT devices, thereby decreasing transmission overhead on wireless channels. Furthermore, Huang et al. [25] focus on designing a lightweight 3D hand-tracking model optimised for edge devices. The study significantly reduces model size while maintaining detection accuracy by optimising model compression and network architecture. Additionally, Li et al. [32] apply the lottery ticket hypothesis to identify and train sparse subnets, thereby reducing the computational load on edge devices and improving the efficiency of computing and communication in FL.

Wu et al. [73] introduce HPC techniques to improve the computational and communication effi-

ciency of neural network accelerators. The paper presents a Genetic Algorithm-based Column Shuffle Remapping scheme to optimise weight distribution in irregularly sparse neural networks, maximising crossbar resource utilisation and compression ratios. The design of a Zero Jump Circuit further speeds up computations by skipping inefficient operations, exemplifying how HPC methodologies can be integrated into EdgeAI to achieve significant improvements in processing speed and energy efficiency. Similarly, Takeuchi [64] highlights the use of CiM and heterogeneous integration to enhance computing efficiency. CiM leverages non-volatile memory technologies to perform massively parallel multiply-accumulate operations, significantly improving throughput and energy efficiency for AI calculations. Integrating CiM with event-driven neuromorphic computing and traditional processors offers an approach to achieving highly power-efficient EdgeAI systems.

FL and distributed strategies are essential for improving efficiency. Xu et al. [75] incorporate HPC techniques into the AEFL framework, leveraging FL, data augmentation, and knowledge distillation to optimise computational processes and minimise communication overhead. Hu et al. [22] address challenges in federated EL by proposing a game-theoretic approach to participation decision-making, ensuring that only devices with sufficient data contributions participate in training. Yang et al. [77] implement a hierarchical tree-based aggregation structure that distributes computing tasks across multiple edge devices, thereby increasing computing efficiency and minimising dependence on a centralised server. Qiu et al. [56] introduce the AI-Chain framework, which uses deep reinforcement learning and distributed consensus protocols to improve computational and communication efficiency, demonstrating that collective reinforcement learning converges faster than traditional approaches. Additionally, Guo et al. [20] employ edge computing servers and FL to distribute computational tasks effectively, reducing latency and bandwidth requirements in VR content delivery. Zeng et al. [82] introduce innovative methods, such as the Sequential-to-Parallel training mode and the Inter-Cluster Grouping algorithm, to optimise computational resources and reduce communication load, applying HPC principles to address constraints in EdgeAI systems.

Zhang et al. [83] explore using HDC as an HPC technique to enhance the efficiency of computing and communication in EdgeAI systems. HDC offers lower computation costs and smaller model sizes than traditional deep neural networks, making it suitable for deployment on resource-constrained edge devices. The paper investigates aggregation schemes in FL that optimise communication between clients and the server, significantly reducing communication overhead by leveraging HDC's computational efficiency.

Advanced optimisation algorithms and mathematical methods are employed to solve complex problems and enhance efficiency. Hudson et al. [26] formulate the joint EI service placement and request scheduling problem as an Integer Linear Program, and develop efficient algorithms such as the Collaborative Greedy Placement to optimise resource utilisation. Huang et al. [24] use Block Coordinate Descent and Successive Convex Approximation algorithms to efficiently solve complex joint optimisation problems involving IRS settings and image sensor parameters. Zhou et al. [89] employ difference convex programming to solve resource allocation optimisation problems, enabling efficient computation of optimal transmission time and power allocation. Lin et al. [41] utilise Lyapunov optimisation and approximate algorithm design to enhance computing efficiency by scheduling DNN layers between CPU and NPU as an online optimisation problem. Similarly, Qin et al. [55] present the EI^3 framework, leveraging optimisation methods like Majorization-Minimisation and Difference-of-Convex programming to jointly optimise learning performance and communication efficiency under time-varying co-channel interference. To improve resource allocation and scheduling, Zhang et al. [85] introduce a depth-first tree-searching algorithm with tree pruning to solve the NP-hard multi-dimensional knapsack problem, thereby optimising batching scheduling and resource allocation for LLM inference. Li, Bi and Wang [34] solve resource allocation problems using the Alternating Direction Multiplier Method, reducing computation time and increasing efficiency. Zhao et al. [87] employ Lyapunov optimisation to efficiently assign verification tasks to heterogeneous edge servers, maximising the number of verified data replicas while adhering to delay constraints, thereby increasing overall efficiency. Wang et al. [70] explore the use of massive MIMO systems to improve communication efficiency, demonstrating that massive MIMO can significantly enhance

learning performance by increasing communication efficiency.

Edge computing architectures and platforms are crucial for managing resources. Psaromanolakis et al. [53] demonstrate that the π-Edge Platform uses a cloud-native architecture based on microservices and container orchestration to manage edge resources effectively. Adopting PaaS and FaaS models ensures scalability and interoperability, while MLOps services introduce automation and intelligent resource management. Benz et al. [6] explore HPC techniques applied in DMAE architectures, detailing how the iDMA architecture leverages parallel processing, distributed computing structures, and optimised data transfer protocols to achieve high bus utilisation and low latency. Additionally, Xia et al. [74] discuss implementing virtualisation technologies and heuristic algorithms based on submodular function maximisation to optimise task scheduling, VM assignment, and resource allocation, minimising system delay and improving efficiency.

AirComp and advanced communication techniques are leveraged to enhance communication efficiency. Liu et al. [45] introduce AirPooling, which employs AirComp to reduce the need for separate computation and communication phases, simplifying data aggregation and significantly reducing latency. Huang et al. [23] incorporate AirComp and joint semantic channel matching, optimising resource allocation by balancing semantic relevance and channel conditions, thereby improving efficiency. Yun et al. [81] explore cooperative DNN inference schemes in which edge devices and servers share the computational workload, optimising the DNN partitioning point and employing error-correcting codes to improve communication reliability and reduce retransmissions.

Deep reinforcement learning and multi-agent systems are applied to improve efficiency. Chen et al. [13] run unique deep reinforcement learning models on each CPU core, acting as EL nodes that optimise local network topology and sharing parameters asynchronously with a global model to accelerate convergence. Liu et al. [43] demonstrate the use of multi-agent deep reinforcement learning for scalable and efficient resource allocation in industrial wireless networks. The system handles complex optimisation tasks and adapts to changing network conditions by enabling multiple agents to learn and make decisions autonomously. Gong et al. [18] utilise Lyapunov stability theory and multi-agent deep federated reinforcement learning to transform complex multi-time-interval optimisation problems into manageable sub-problems, enabling efficient computation offloading and resource allocation.

Advanced algorithms and ML models contribute to efficiency improvements. Wang et al. [71] focus on using GNNs to optimise resource allocation, aiming to reduce computational complexity and increase scalability by efficiently mapping complex channel information to optimal system configurations. Kim et al. [30] employ ViT and masked autoencoders to efficiently process high-dimensional data in channel estimation, optimising resource use in communication systems. Minh, Mai and Minh [47] demonstrate the implementation of deep learning models such as YOLOv4 and MobileNet-SSD on devices with GPU acceleration, using parallel processing and model optimisation to improve computational performance on resource-limited devices.

Tuli, Casale and Jennings [67] employ advanced neural network architectures, such as Transformers, to model temporal trends and closed-graph convolutional networks to capture spatial correlations in the system topology. In addition, SimTune uses gradient-based Monte Carlo search to tune simulator parameters efficiently. These HPC methods enable fast, accurate predictions of system behaviour, enabling more efficient task scheduling and resource management. Consequently, this leads to reduced response times, lower energy consumption and better scalability in EdgeAI systems.

Security and robustness considerations also affect efficiency. Chen et al. [12] present FedEqual, a defence strategy in FL that scales the gradients of each local model to a common L2 norm, improving robustness against model poisoning attacks while maintaining performance, thereby enhancing computing efficiency. Abdel-Basset, Moustafa and Hawash [1] use blockchain technology for distributed aggregation and validation in the Fed-Trust framework, reducing dependence on centralised servers and improving resource efficiency while implementing group normalisation to decrease computational complexity.

Collectively, these studies showcase a diverse array of HPC techniques used to improve the efficiency of computing and communication in EdgeAI systems. These approaches address the inherent chal-

lenges of resource constraints, latency, and scalability by leveraging specialised hardware accelerators and advanced optimisation algorithms, as well as implementing FL and AirComp. By integrating these techniques, researchers demonstrate significant improvements in processing speed, energy efficiency, and overall system performance, thereby advancing EdgeAI applications. To conclude, the application of High Performance Computing techniques in EdgeAI signifies a strategic shift. It moves beyond simple hardware upgrades to a fundamental restructuring of where processing occurs. By leveraging parallelism and specialised accelerators locally, these strategies effectively decouple system performance from network availability. This reinforces the central thesis that robust Computing optimisation is often the most reliable method to guarantee Communication efficiency in distributed architectures.

Figure 7 summarises HPC techniques applied to EdgeAI, organised into five categories: hardware accelerators, compiler optimisations, model compression, parallelism and networking.

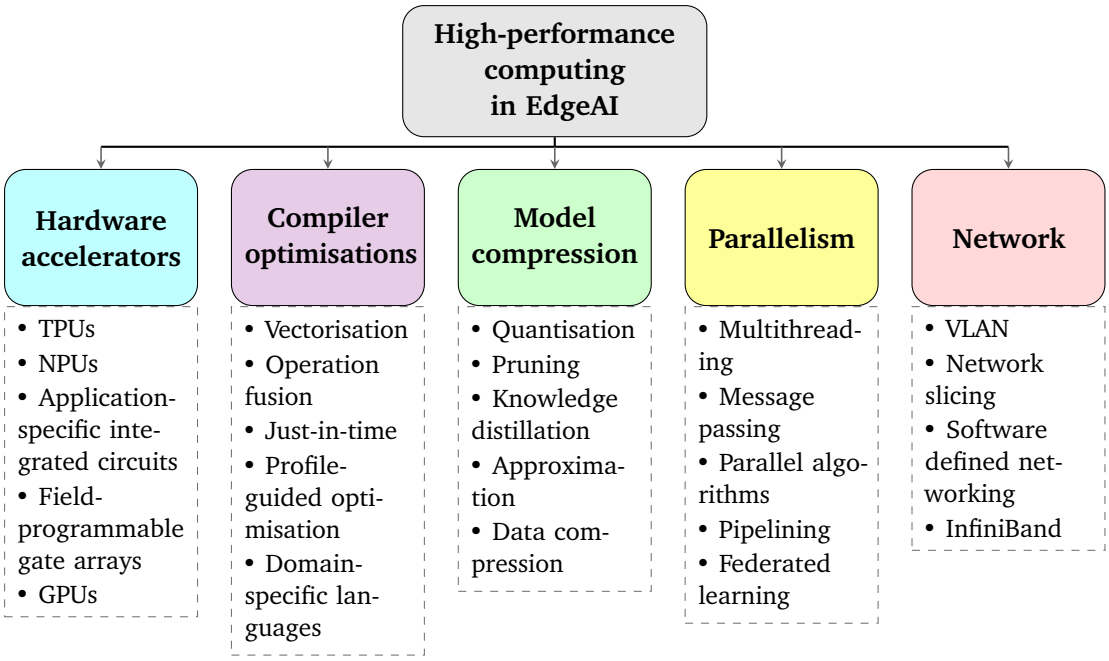

Figure 7: HPC techniques resume.

4.6. SQ4: What are the main policies and penalties when addressing quality of service in the EdgeAI context?

QoS in EdgeAI systems is a critical factor that directly impacts user experience and system efficiency. Various policies and penalties have been proposed to manage and optimise QoS, often involving trade-offs between metrics such as accuracy, latency, power consumption, and resource utilisation. For instance, Hudson et al. [26] define QoS in terms of two key metrics: the accuracy of AI service implementations and the end-user delay. The paper discusses the inherent trade-offs between accuracy and latency across different AI models deployed at the edge, noting that more complex models offer higher accuracy but incur greater latency. To manage these trade-offs, the authors propose policies for service placement and request scheduling that maximise total QoS. They incorporate penalties into the QoS calculation for services that do not meet user-defined accuracy or delay thresholds, thereby directly addressing how penalties can be applied within QoS policies. By formulating QoS as an optimisation problem, the paper provides strategies for balancing resource constraints with user QoS requirements in the EdgeAI context.

Similarly, Lin et al. [41] introduce policies to manage power consumption while maintaining QoS in EdgeAI systems. The paper implements a long-term time-averaged power cap as a policy to

maintain hardware reliability and lifespan. The scheduling framework aims to enforce this power cap while minimising inference accuracy loss and meeting real-time performance constraints, such as processing each video frame before the next one arrives. The penalties involved include potential increases in inference accuracy loss when scheduling decisions favour power cap enforcement or performance constraints. The framework balances these policies and penalties by employing Lyapunov optimisation, providing theoretical guarantees on the trade-offs between accuracy loss and power consumption. The control parameter V allows system designers to adjust the relative importance of accuracy versus power constraints, effectively managing QoS in the EdgeAI context by tuning system performance according to application requirements.

In the realm of FL, managing QoS involves addressing issues such as communication efficiency, learning accuracy, and the presence of stragglers. Guo et al. [19] focus on the straggler problem in FL, which can significantly impact learning performance. To mitigate the effects of stragglers, the article imposes a maximum delay constraint per communication round, ensuring that the completion of global aggregation does not exceed a predefined threshold. While it does not specify explicit QoS-related policies or penalties, addressing the straggler problem implicitly highlights its importance as a key policy in edge FL. Complementing this, Qin et al. [55] propose resource allocation strategies that balance communication efficiency and learning accuracy. It considers the communication requirements of IIoT devices, such as guaranteeing minimum achievable rates and satisfying QoS constraints. The low-cost and balance-participating algorithm (LCBPA) minimises the difference between the overall classification error and the minimum achievable rate while ensuring that communication requirements are met. This involves formulating constraints to guarantee communication QoS and demonstrates how policy decisions in power allocation can impact QoS in EdgeAI systems.

Reinforcement learning approaches also contribute to QoS management. Hlophe, Awoyemi and Maharaj [21] implicitly incorporate QoS policies through their focus on user satisfaction and minimising packet loss. It employs subjective metrics, such as interference and congestion rates, to evaluate device satisfaction, thereby effectively creating internal policies that prioritise actions leading to successful offloading and high user satisfaction. The reward function within the reinforcement learning model serves as a mechanism to enforce these policies by rewarding actions that reduce packet loss and congestion while penalising those that do not meet QoS requirements. Although the article does not detail specific external policies or penalties, the intrinsic reward-based approach establishes a framework for maintaining QoS standards by incentivising desirable behaviours and discouraging actions that degrade the QoS.

Moreover, QoS management can involve incentive mechanisms to ensure the system's reliability and trustworthiness. Abdel-Basset, Moustafa and Hawash [1] discuss implementing an incentive mechanism within the Fed-Trust framework to maintain QoS. It introduces policies that reward participants based on their contributions and reputation scores, encouraging honest participation in the network. Malicious behaviour, such as uploading harmful model updates, is penalised by reducing the participant's reputation and reward values. This system of rewards and penalties helps ensure the EdgeAI system's reliability and trustworthiness by preventing malicious activity that can degrade performance.

Finally, proactive system adaptations and monitoring help maintain QoS. Psaromanolakis et al. [53] address QoS management through MLOps services, incorporating proactive scaling and adaptation mechanisms that use predefined thresholds to maintain QoS levels. By continuously monitoring resource utilisation and employing ML models to predict potential QoS degradations, the platform applies certain policies to guarantee QoS. While detailed policies and penalties are not specified, the approach underscores the importance of adaptive mechanisms in maintaining QoS in EdgeAI contexts.

These studies highlight the multifaceted strategies employed to address QoS in EdgeAI systems. Policies and penalties are implemented through various mechanisms, including optimisation frameworks, resource allocation strategies, reinforcement learning reward functions, incentive mechanisms, and proactive system adaptations. These approaches aim to balance competing demands such as accuracy, latency, power consumption, and resource utilisation, ensuring that QoS requirements are

met while optimising overall system performance. In essence, QoS policies serve as the arbitration mechanism that governs the interplay between computing, storage, and communication. Rather than regulating these domains in isolation, a robust QoS framework enforces a unified resource strategy. It determines when to sacrifice model accuracy to meet latency constraints or when to expend energy on local storage to avoid communication delays. This establishes QoS not merely as a performance metric but as the decisive factor that orchestrates the cross-pillar trade-offs essential for reliable EdgeAI.

5. Discussion and research opportunities

In this section, we synthesise insights from our comprehensive survey of optimisations for computing, storage, and communication in EdgeAI systems. We propose a detailed taxonomy that categorises optimisation strategies, providing a structured framework for understanding the current landscape. This discussion highlights the interrelationships among different approaches and identifies key opportunities to enhance system performance, efficiency, and scalability. Additionally, we examine the existing gaps that hinder the optimal deployment of EdgeAI solutions, highlighting areas that warrant further research and development.

5.1. Taxonomy

In our proposed taxonomy for EdgeAI Optimisation, illustrated in figure 8, we summarise the key achievements identified in our survey and categorise them into four primary domains: *Computing Optimisation*, *Storage Optimisation*, *Communication Optimisation*, and *Cross-cutting Optimisations*. Each of these categories addresses a distinct pillar of improvement, including strategies that improve system performance, reduce latency, and optimise resource usage at the edge.

Although these categories are presented independently, they are intrinsically interconnected. For instance, compressing a transmitted message not only reduces bandwidth consumption – improving communication efficiency – but also minimises the amount of storage space required for that message once it reaches its destination. By capturing these multifaceted relationships, the taxonomy provides a holistic perspective of the optimisation approaches available for achieving robust and efficient EdgeAI systems. This structured taxonomy specifically addresses the gap identified in the related work (section 2). While previous surveys, such as Shi et al. [61], focus predominantly on communication efficiency, or Liang et al. [39] concentrate on model compression, our framework integrates these domains. By categorising optimisations not just by their primary function but by their cross-cutting impact, our taxonomy provides a more actionable map for system architects. It highlights that strategies like model quantisation serve simultaneously as storage optimisations and bandwidth savers, offering a dual benefit often overlooked in single-domain reviews.

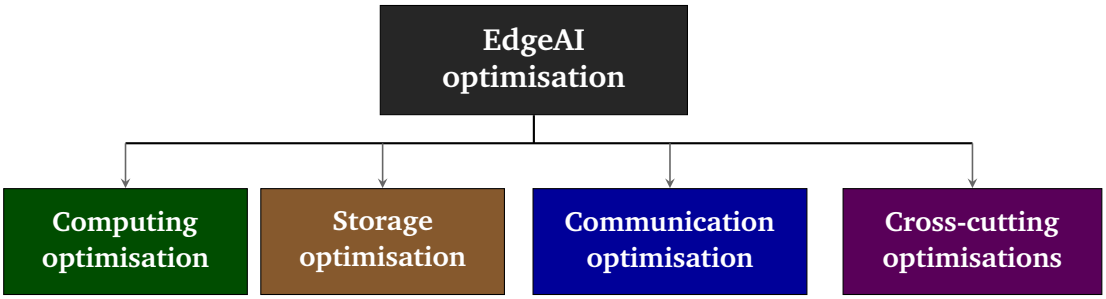

Figure 8: Taxonomy primary categories.

Computational optimisation in EdgeAI encompasses various strategies to enhance performance, energy efficiency, and scalability across diverse hardware and software levels. These strategies include using specialised accelerators, heterogeneous computing architectures that integrate CPUs and

specialised units, and advanced model optimisation techniques – such as compression, quantisation, and sparse representations – to reduce the computational footprint of AI models. Additionally, distributed learning approaches, like FL and three-tier device-edge-cloud architectures, help maintain data privacy while balancing processing loads across the network. Figure 9 organises these concepts into a taxonomy, providing an overview of computing optimisation strategies in the context of EdgeAI.

Storage optimisation in EdgeAI involves hardware and software innovations that ensure efficient data handling close to the data source. Advancements in non-volatile memory technologies – such as ReRAM, PRAM, FeFET, and Flash – offer improved performance and endurance. Complementing these technologies, data reduction techniques, including pruning, compression, and quantisation, significantly decrease storage requirements and data traffic, while edge caching strategies and distributed storage management solutions (potentially supported by blockchain) maintain the integrity, security, and accessibility of information. As illustrated in figure 10, these elements form a taxonomy of storage optimisation methods tailored for EdgeAI environments.

Efficient communication mechanisms ensure that EdgeAI systems meet stringent latency, bandwidth, and reliability requirements. Techniques such as bandwidth allocation, power control, and dynamic offloading help optimise data flows while emerging technologies – including RIS and semantic communication – enable more intelligent data exchange. Security and privacy measures, supported by blockchain integration, ensure trustworthy communication. Drawing together these multifaceted aspects, figure 11 presents a taxonomy that captures the range of communication optimisation strategies supporting next-generation EdgeAI deployments.

Ensuring the seamless integration of computing, storage, and communication optimisations requires a holistic approach that addresses challenges across multiple layers simultaneously. Such cross-cutting optimisations involve co-design strategies where hardware, software, and networking solutions are developed in tandem. For instance, selecting a specific hardware accelerator (computing) directly determines the available memory-compression formats (storage), which, in turn, define the required transmission protocols (communication). Techniques that unify workload orchestration and resource allocation enable adaptable systems by treating these resources as interchangeable variables in a global optimisation function. Additionally, robust security and trust frameworks must be integrated at every stage to reinforce system integrity without compromising the efficiency gains achieved by other pillars. Figure 12 presents the taxonomy of cross-cutting optimisations, illustrating how multifaceted approaches harmonise the EdgeAI ecosystem.

The taxonomy proposed in this paper differs significantly from the structures presented by the TRs reviewed. In Surianarayanan et al. [63], the classification focuses only on model optimisation (pruning, quantisation, distillation) and does not address network or memory policies. Shi et al. [61] provide an exhaustive overview of communication efficiency but treat computing and storage as marginal topics. The survey by Duan et al. [16] introduces a holistic framework for distributed AI on EECC architectures. However, the categories are derived from the training/inference cycle rather than the optimisation target. Boucetta et al. [8] restrict themselves to satellite image streams and model only communication-processing challenges; Liang et al. [39] focus on model summarisation, while Khouas et al. [29] deal mostly with data orchestration strategies. Thus, none of these TRs simultaneously maps the three pillars or explains techniques that act across them; our taxonomy fills this gap by offering a single framework that allows us to compare - and eventually compose - co-dependent optimisations for edge AI.

A critical comparison of the proposed taxonomy indicates that the most suitable optimisation strategy depends primarily on the dominant system bottleneck. Storage and computing optimisations, such as network pruning, quantisation, operator fusion, and lightweight model redesign, are more appropriate for edge nodes with severe thermal, memory, or storage limitations, especially when network connectivity is sufficiently stable, and communication is not the main source of degradation. In these conditions, reducing model size and inference costs improves deployability, runtime stability, and energy efficiency in a direct, practical way.

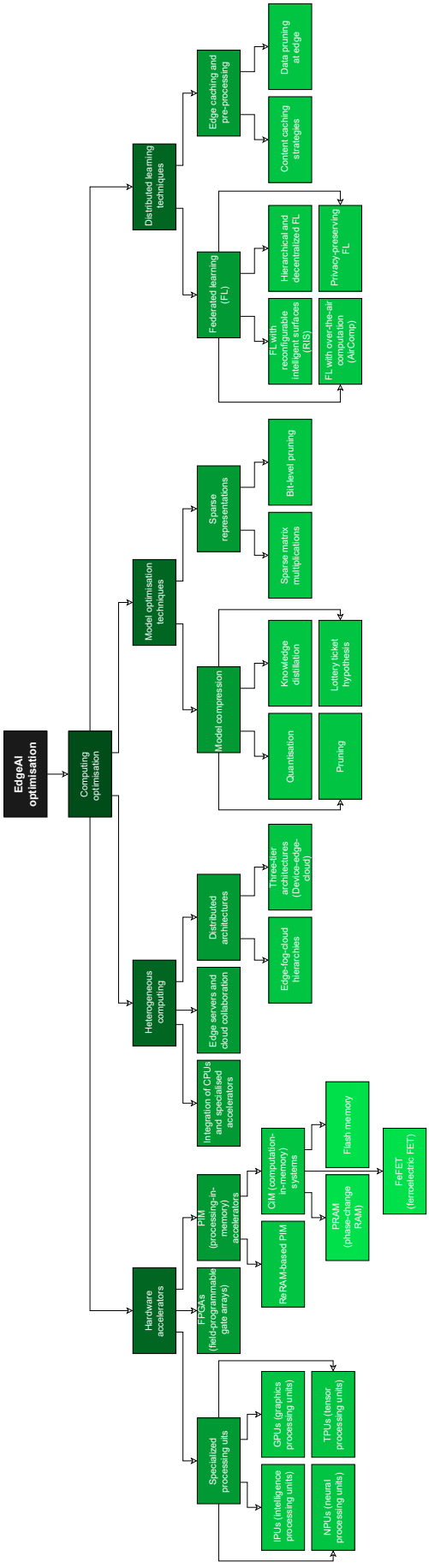

Figure 9: Taxonomy of computing optimization strategies for EdgeAI, encompassing hardware accelerators, heterogeneous architectures, model optimization techniques, and distributed learning approaches.

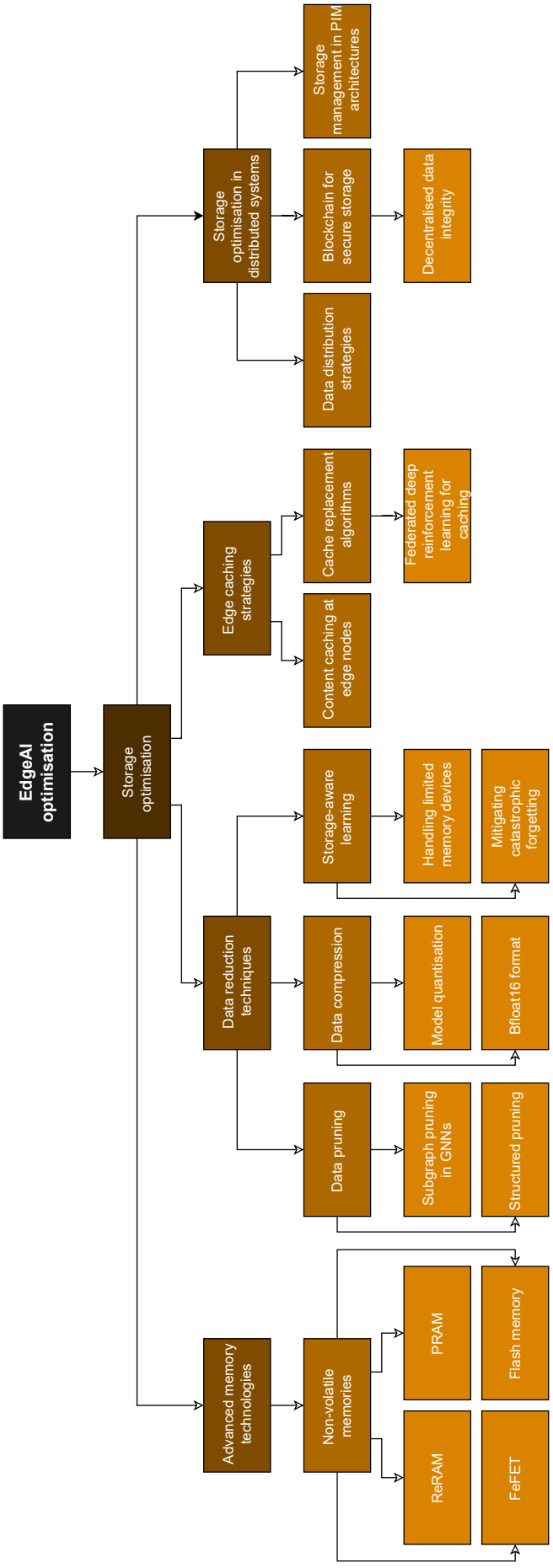

Figure 10: Taxonomy of storage optimization strategies in EdgeAI, including advanced non-volatile memory technologies, data reduction techniques, edge caching, and distributed storage management.

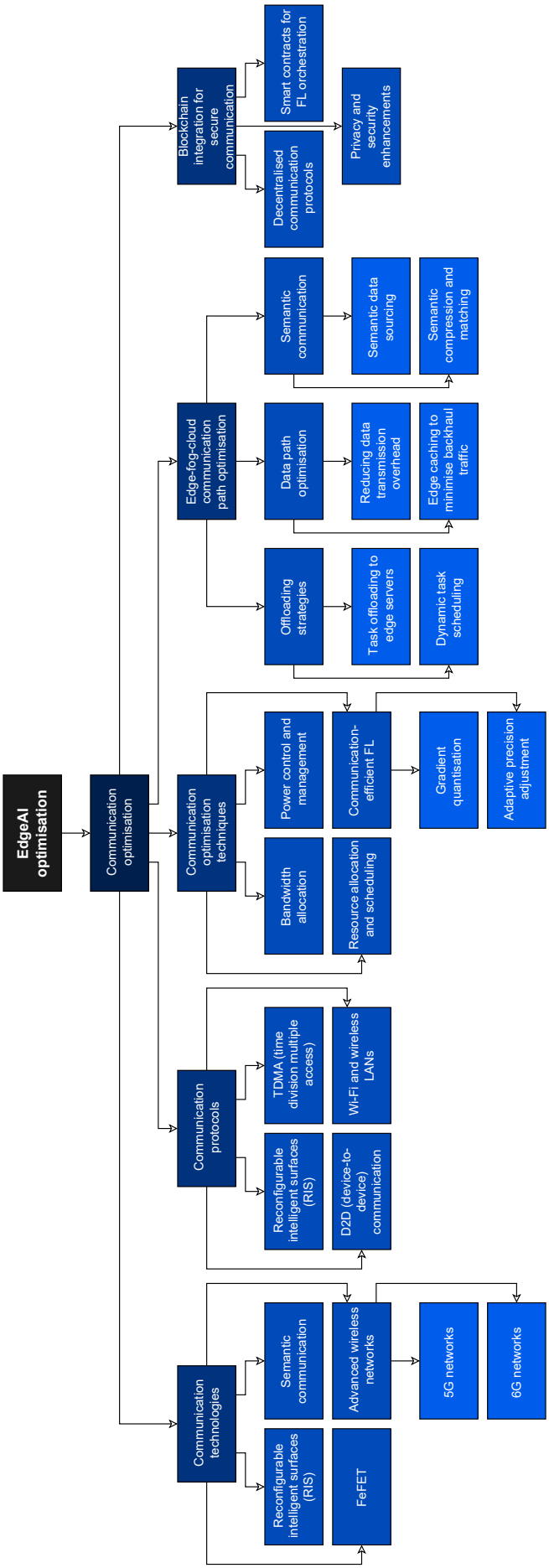

Figure 11: Taxonomy of communication optimization strategies in EdgeAI, incorporating emerging communication technologies, protocols, resource allocation techniques, offloading strategies, and blockchain-based security measures.

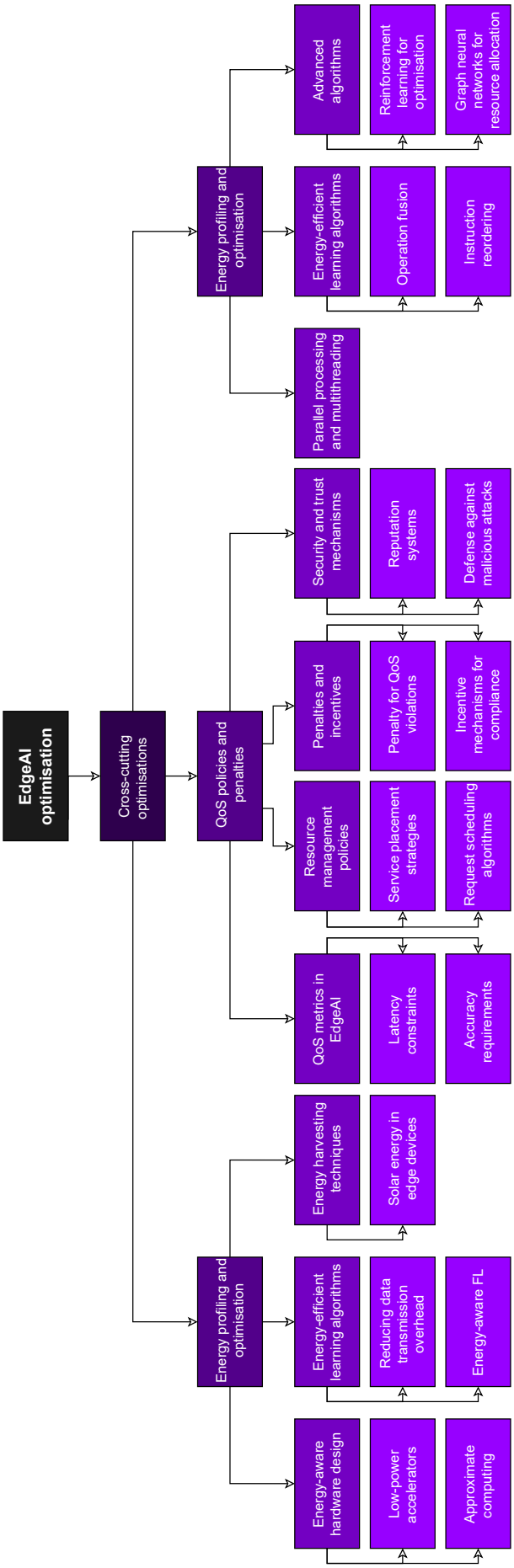

Figure 12: Taxonomy of cross-cutting optimizations in EdgeAI, encompassing co-design strategies, unified orchestration methods, advanced analytical techniques, and integrated security and trust frameworks.

By contrast, communication-oriented strategies, such as Federated Learning, semantic compression, adaptive offloading, and bandwidth-aware transmission, are more suitable in environments where bandwidth is intermittent, costly, or highly variable, provided the participating nodes still have sufficient local computational capacity to sustain partial processing and coordination. From a design perspective, systems limited mainly by on-device resource scarcity should prioritise storage and computing efficiency. In contrast, systems limited mainly by network instability should prioritise communication efficiency, since each category addresses a different operational bottleneck in the EdgeAI continuum.

Table 5 presents representative scenarios illustrating how the categories defined in our taxonomy drive specific technological choices. For example, in the "Remote Patient Monitoring" scenario, the taxonomy's "Data Reduction" category (storage) guides the decision to use model quantisation. This choice effectively reduces the payload size, which then allows the "Bandwidth management" strategy (communication) to utilise lightweight protocols like MQTT even under poor network conditions. By mapping the problem through the lens of computing, communication, and storage simultaneously, the table demonstrates that performance gains arise not from isolated decisions, but from the coordinated selection of technologies across all three pillars.

Table 5
Illustrative use cases and their optimisations.

Scenario	Computation	Communication	Storage	Key benefit
Remote patient monitoring	Quantisation of on-device NEWS2 models	MQTT + adaptive compression	24h local cache + blockchain for auditing	Reduces latency and traffic while maintaining data integrity
Collaborative agricultural drones	Lightweight FPGA for YOLO-tiny	AirComp for merging NDVI maps between drones	Buffer in ReRAM PIM for offline flights	Real-time analysis with extended autonomy
Factory 4.0: fault prediction	Federated incremental learning in gateways	5G URRLC with network slicing	Adaptive caching by DRL for vibration logs	Predictive maintenance sub-50 ms

5.2. Opportunities

Exploring EdgeAI architectures unveils numerous opportunities to enhance system performance, efficiency, and scalability. A significant opportunity lies in optimising edge caching strategies and data path enhancements between the edge, fog, and cloud layers. Systems can reduce latency and bandwidth consumption by intelligently caching frequently accessed data and refining communication models – such as implementing AirComp and adaptive data compression. These optimisations enable real-time processing in edge environments, which is crucial for applications that require immediate data analysis and response.

Integrating advanced ML models, particularly TinyML [65], offers substantial potential for performance improvements at the edge. TinyML focuses on implementing ML algorithms on microcontrollers and other resource-constrained devices. Computational and storage requirements are reduced by developing lightweight AI models using techniques such as model pruning, quantisation, and knowledge distillation. Furthermore, establishing standardised API systems for TinyML can create a unified framework for deploying ML at the edge. Standardisation enables modifications to the engine or other parameters while maintaining compatibility and interoperability across different devices and applications. Such an approach streamlines development and facilitates the widespread adoption of EdgeAI technologies, enabling deployment on devices with limited resources and unlocking new applications in areas such as environmental monitoring, healthcare, and predictive maintenance, where low power consumption and real-time processing are essential.

Although our taxonomy adopts an infrastructure-agnostic perspective, mature application domains such as continuous healthcare monitoring clearly demonstrate the practical value of these combined optimisations. In wearable health systems, local processing helps preserve privacy and enables immediate interpretation of physiological signals, while data compression reduces transmission overhead and energy consumption, extending battery life. At the same time, efficient communication mechanisms support the timely delivery of critical alerts when abnormal patterns are detected. These combined strategies have already been empirically validated as effective for addressing the latency and power constraints that remain central bottlenecks in wearable EdgeAI deployments.

Enhancing development boards like Arduino, Raspberry Pi, and Banana Pi to be more reliable and capable of AI processing presents a significant opportunity. While these platforms offer accessible, cost-effective hardware solutions, their capabilities can be improved to support AI workloads better. Upgrading these boards with more powerful processors, increased memory capacity, and the integration of specialised AI accelerators can enable them to handle complex ML tasks more efficiently. Optimising their software frameworks for AI applications and enhancing their energy management features can make them more reliable for continuous operation in edge environments. Such improvements would extend their applicability beyond prototyping to more demanding industrial and commercial EdgeAI deployments.

In addition to other boards, the use of System-on-a-Chip (SoC) technology represents a significant opportunity for EdgeAI systems. SoCs integrate all the components of a computer and other electronic systems into a single chip, offering high performance in a compact form factor. By leveraging SoCs, developers can create edge devices that are both powerful and energy-efficient, capable of handling complex processing tasks while maintaining low power consumption. Such advancements are particularly beneficial for applications requiring real-time data processing and analysis at the edge.

Enhancing energy profiling and management tools explicitly designed for edge devices presents a promising opportunity. Detailed energy profiling provides insights into consumption patterns, enabling the development of effective energy-saving strategies. Integrating these tools with resource management systems supports dynamic adjustments to operational parameters, optimising energy usage without compromising performance. Such advancements are especially crucial for battery-powered edge devices, as efficient energy management extends their operational life and enhances reliability.

Adapting HPC techniques for edge environments holds untapped potential. Significant performance gains can be achieved by bringing HPC methodologies, such as parallel processing algorithms and hardware acceleration technologies, to the edge. Overcoming the challenges associated with this adaptation can yield robust EdgeAI systems capable of handling complex tasks efficiently, opening new possibilities in fields such as autonomous vehicles, smart cities, and industrial automation.

Figure 13 summarises the main gaps and opportunities in the EdgeAI field, highlighting areas such as optimisation of edge caching and communication, advances in resource allocation algorithms, integration of TinyML with standardised Application Programming Interfaces, improvements in development boards for AI processing, use of SoCs, improvement of energy management and adaptation of HPC techniques. Focusing on these opportunities can significantly boost the capabilities and effectiveness of distributed EdgeAI architectures.

6. Limitations and future work

Our study identified several key limitations:

- *Gap between theoretical advancements and practical industry applications*: There is a disconnect between academic research and the real-world implementation of EdgeAI systems. Theoretical models often do not account for the practical challenges industry professionals face. Future work should bridge this gap by engaging directly with industry professionals through collaborations and partnerships. Incorporating case studies and empirical data from commercial deployments can enhance the practical viability and adoption of optimisation strategies.

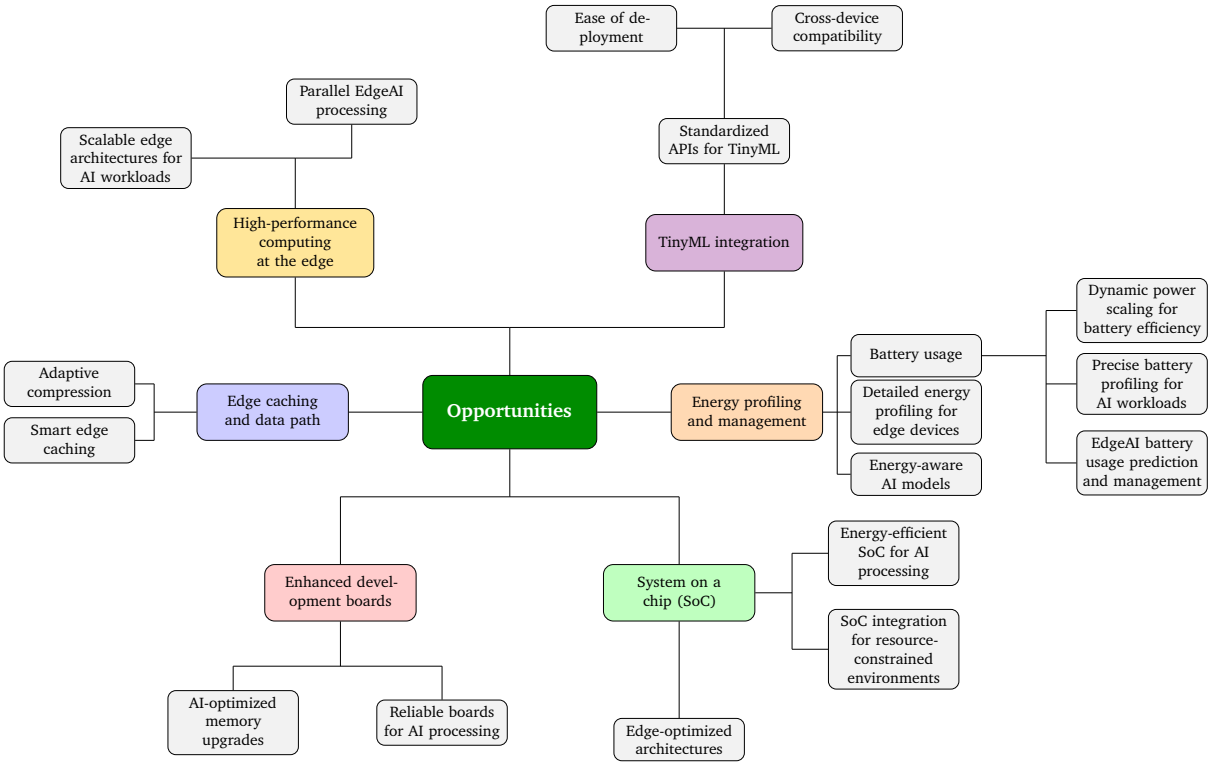

Figure 13: Open gaps.

- *Insufficient integration of economic and regulatory considerations*: Current research tends to overlook factors such as cost-effectiveness, return on investment, and compliance with regional regulations, which are crucial for the deployment and optimisation of EdgeAI systems. Future studies should integrate economic and regulatory factors into EdgeAI research, examining cost-benefit analyses, return on investment, and regulatory compliance to offer recommendations that address both technical challenges and critical economic and legal considerations.
- *Search strategy trade-off*: The broad search string employed to cover the intersection of computing, storage, and communication prioritised recall over precision. While this strategy was necessary to build a cross-domain taxonomy, it is possible that specific niche studies were not captured. However, the selected articles serve as representative archetypes for the discussed optimisation categories.
- *Standardisation of evaluation metrics*: The current literature employs a heterogeneous mix of metrics, ranging from objective parameters (latency, throughput, energy joules) to subjective quality of experience (QoE) indicators. Future surveys should focus on compiling and categorising these metrics to establish a standardised benchmarking framework for EdgeAI optimisations.

7. Conclusion

In this survey, we provide a comprehensive, holistic analysis of EdgeAI technologies, focusing on critical optimisations in communication, storage, and computing within distributed system architectures. By meticulously reviewing existing research, we identified key strategies and technologies that enable the efficient deployment of ML tasks on edge devices, addressing the inherent challenges posed by limited resources and energy constraints.

Our work fills a significant gap in the literature by offering an integrated perspective on EdgeAI optimisations. Unlike prior studies that often focus on specific aspects, we synthesised insights across

the communication, storage, and computing domains, highlighting their interdependencies and uncovering synergistic optimisations that can substantially enhance performance, reduce latency, and improve energy efficiency in edge-fog-cloud ecosystems. By exploring key trends and techniques such as specialised hardware accelerators, FL enhancements, model compression methods, advanced communication protocols, and intelligent resource allocation algorithms, we provide valuable guidance for researchers and practitioners aiming to advance the field of EdgeAI.

Moreover, our proposed taxonomy provides a structured framework for understanding diverse optimisation strategies and their interrelationships, serving as a valuable resource for designing and deploying intelligent edge systems. We also discussed gaps and opportunities in the field, emphasising the need for standardised frameworks, advanced ML models tailored for edge environments, enhanced security mechanisms, and comprehensive QoS management systems. Addressing these gaps is essential for the widespread adoption and scalability of EdgeAI solutions.

Our survey will serve as a reference for ongoing developments in EdgeAI technologies. By shedding light on the complexities and potential of integrated optimisation, we hope to inspire future research that builds on our findings. The advancements in EdgeAI promise to transform how intelligent services are delivered, bringing enhanced capabilities closer to users and enabling a new generation of applications across various domains.

Author contributions

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

Funding

This research received no external funding.

Data availability statement

No new data were created or analysed during this study. Data sharing is not applicable.

Conflicts of interest

The authors declare no conflict of interest.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] Abdel-Basset, M., Moustafa, N. and Hawash, H., 2022. Privacy-Preserved Cyberattack Detection in Industrial Edge of Things (IEoT): A Blockchain-Orchestrated Federated Learning Approach. *IEEE Transactions on Industrial Informatics*, 18(11), pp.7920–7934. Available from: https://doi.org/10.1109/TII.2022.3167663.

[2] Al Ridhawi, I., Bouachir, O., Aloqaily, M. and Boukerche, A., 2021. Design Guidelines for Cooperative UAV-supported Services and Applications. *Acm computing surveys*, 54(9), p.185. Available from: https://doi.org/10.1145/3467964.

[3] Ang, P.L., Rana, M.E. and Hameed, V.A., 2023. Revolutionizing Finance: The Transformative Impact of Cloud Computing in Finance Shared Service Center (FSSC). *2023 IEEE 21st Student Conference on Research and Development (SCOReD)*. pp.482–488. Available from: https://doi.org/10.1109/SCOReD60679.2023.10563756.

[4] Askarizadeh, M., Morsali, A. and Nguyen, K.K., 2025. Resource-Constrained Multisource Instance-Based Transfer Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1), pp.1029–1043. Available from: https://doi.org/10.1109/TNNLS.2023.3327248.

[5] Babaei, P., 2024. Convergence of Deep Learning and Edge Computing using Model Optimization. *2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP)*. pp.1–6. Available from: https://doi.org/10.1109/MVIP62238.2024.10491145.

[6] Benz, T., Rogenmoser, M., Scheffler, P., Riedel, S., Ottaviano, A., Kurth, A., Hoefler, T. and Benini, L., 2024. A High-Performance, Energy-Efficient Modular DMA Engine Architecture. *IEEE Transactions on Computers*, 73(1), pp.263–277. Available from: https://doi.org/10.1109/TC.2023.3329930.

[7] Binucci, F., Banelli, P., Di Lorenzo, P. and Barbarossa, S., 2023. Analog versus Digital Pulse Amplitude Modulation for Goal-Oriented Wireless Communications. *2023 31st European Signal Processing Conference (EUSIPCO)*. pp.1415–1419. Available from: https://doi.org/10.23919/EUSIPCO58844.2023.10289715.

[8] Boucetta, A.Y., Baziz, M., Hamdad, L. and Allal, I., 2024. Optimizing for Edge-AI Based Satellite Image Processing: A Survey of Techniques. *2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*. pp.83–87. Available from: https://doi.org/10.1109/M2GARSS57310.2024.10537575.

[9] Chan, Y.W., Fathoni, H., Yen, H.Y. and Yang, C.T., 2022. Implementation of a Cluster-Based Heterogeneous Edge Computing System for Resource Monitoring and Performance Evaluation. *IEEE Access*, 10, pp.38458–38471. Available from: https://doi.org/10.1109/ACCESS.2022.3166154.

[10] Chen, C., Jiang, B., Liu, S., Li, C., Wu, C. and Yin, R., 2023. Efficient Federated Learning using Random Pruning in Resource-Constrained Edge Intelligence Networks. *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. pp.5244–5249. Available from: https://doi.org/10.1109/GLOBECOM54140.2023.10437051.

[11] Chen, C., Jiang, B., Liu, S., Li, C., Wu, C. and Yin, R., 2024. Efficient Federated Learning in Resource-Constrained Edge Intelligence Networks Using Model Compression. *IEEE Transactions on Vehicular Technology*, 73(2), pp.2643–2655. Available from: https://doi.org/10.1109/TVT.2023.3318080.

[12] Chen, L.Y., Chiu, T.C., Pang, A.C. and Cheng, L.C., 2021. FedEqual: Defending Model Poisoning Attacks in Heterogeneous Federated Learning. *2021 IEEE Global Communications Conference (GLOBECOM)*. pp.1–6. Available from: https://doi.org/10.1109/GLOBECOM46510.2021.9685082.

[13] Chen, N., Qiu, T., Zhao, L., Zhou, X. and Ning, H., 2021. Edge Intelligent Networking Optimization for Internet of Things in Smart City. *IEEE Wireless Communications*, 28(2), pp.26–31. Available from: https://doi.org/10.1109/MWC.001.2000243.

[14] Chi, X., Han, S., Xu, X., Li, L., Wang, H., Qin, X., Jin, L. and Zhang, P., 2024. Source Value-Based Resource Allocation in Task-Oriented Communications. *IEEE Internet of Things Journal*, 11(24), pp.39395–39408. Available from: https://doi.org/10.1109/JIOT.2024.3430905.

[15] Desnos, K., Bourgoin, T., Dardaillon, M., Sourbier, N., Gesny, O. and Pelcat, M., 2022. Ultra-Fast Machine Learning Inference through C Code Generation for Tangled Program Graphs. *2022 IEEE Workshop on Signal Processing Systems (SiPS)*. pp.1–6. Available from: https://doi.org/10.1109/SiPS55645.2022.9919237.

[16] Duan, S., Wang, D., Ren, J., Lyu, F., Zhang, Y., Wu, H. and Shen, X., 2023. Distributed Artificial Intelligence Empowered by End-Edge-Cloud Computing: A Survey. *IEEE Communications Surveys & Tutorials*, 25(1), pp.591–624. Available from: https://doi.org/10.1109/COMST.2022.3218527.

[17] Gong, X., Zhang, J., Zhang, Y., Tan, Y. and Song, J., 2022. Inverse-projection Transformation Based Feature Learning Method for Edge Intelligence Approaches Under Different Working Conditions. *2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)*. pp.258–265. Available from: https://doi.org/10.1109/MLCCIM55934.2022.00051.

[18] Gong, Y., Yao, H., Liu, X. and Nallanathan, A., 2023. Privacy-Assisted Computation Offloading Schemes for Satellite-Ground Digital Twin Networks. *ICC 2023 - IEEE International Conference on Communications*. pp.723–728. Available from: https://doi.org/10.1109/ICC45041.2023.10279144.

[19] Guo, K., Chen, Z., Yang, H.H. and Quek, T.Q.S., 2022. Dynamic Scheduling for Heterogeneous Federated Learning in Private 5G Edge Networks. *IEEE Journal of Selected Topics in Signal Processing*, 16(1), pp.26–40. Available from: https://doi.org/10.1109/JSTSP.2021.3126174.

[20] Guo, Y., Qin, Z., Tao, X. and Li, G.Y., 2024. Federated Multi-View Synthesizing for Metaverse. *IEEE Journal on Selected Areas in Communications*, 42(4), pp.867–879. Available from: https://doi.org/10.1109/JSAC.2023.3345427.

[21] Hlophe, M.C., Awoyemi, B.S. and Maharaj, B.T., 2023. Deep Q-learning Network Solution for Subjective Computational Task Scheduling in Edge Platforms. *2023 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. pp.132–137. Available from: https://doi.org/10.1109/ANTS59832.2023.10469441.

[22] Hu, Q., Li, F., Zou, X. and Xiao, Y., 2020. Correlated Participation Decision Making for Federated Edge Learning. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. pp.1–6. Available from: https://doi.org/10.1109/GLOBECOM42002.2020.9321981.

[23] Huang, K., Lan, Q., Liu, Z. and Yang, L., 2023. Semantic Data Sourcing for 6G Edge Intelligence. *IEEE Communications Magazine*, 61(12), pp.70–76. Available from: https://doi.org/10.1109/MCOM.001.2200962.

[24] Huang, N., Dou, C., Wu, Y., Qian, L., Zhou, S. and Lu, R., 2025. Image Analysis Oriented Integrated Sensing and Communication via Intelligent Reflecting Surface. *IEEE Transactions on Cognitive Communications and Networking*, 11(1), pp.274–287. Available from: https://doi.org/10.1109/TCCN.2024.3414393.

[25] Huang, Y., Luo, C., Feng, X., Yang, Z., Zhang, J. and Li, J., 2022. LigNet: Lightweight Hand Tracking for Edge Intelligence. *2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. pp.636–642. Available from: https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics55523.2022.00036.

[26] Hudson, N., Khamfroush, H., Baughman, M., Lucani, D.E., Chard, K. and Foster, I., 2024. QoS-aware edge AI placement and scheduling with multiple implementations in FaaS-based edge computing. *Future Generation Computer Systems*, 157, pp.250–263. Available from: https://doi.org/10.1016/j.future.2024.03.035.

[27] Iftikhar, S., Gill, S.S., Song, C., Xu, M., Aslanpour, M.S., Toosi, A.N., Du, J., Wu, H., Ghosh, S., Chowdhury, D., Golec, M., Kumar, M., Abdelmoniem, A.M., Cuadrado, F., Varghese, B., Rana, O., Dustdar, S. and Uhlig, S., 2023. AI-based fog and edge computing: A systematic review, taxonomy and future directions. *Internet of Things*, 21, p.100674. Available from: https://doi.org/10.1016/j.iot.2022.100674.

[28] Ji, Z. and Qin, Z., 2023. Energy-Efficient Task Offloading for Semantic-Aware Networks. *ICC 2023 - IEEE International Conference on Communications*. pp.3584–3589. Available from: https://doi.org/10.1109/ICC45041.2023.10279646.

[29] Khouas, A.R., Bouadjenek, M.R., Hacid, H. and Aryal, S., 2024. Training machine learning models at the edge: A survey. *arxiv*. Available from: https://doi.org/10.48550/arXiv.2403.02619.

[30] Kim, K., Tun, Y.K., Shirajum Munir, M., Saad, W. and Hong, C.S., 2024. Pilot Optimization and Channel Estimation Scheme for Semantic Communication: A Framework for Edge Intelligence.

NOMS 2024-2024 IEEE Network Operations and Management Symposium. pp.1–7. Available from: https://doi.org/10.1109/NOMS59830.2024.10575676.

[31] Landsmeer, L.P., Engelen, M.C., Miedema, R. and Strydis, C., 2024. Tricking AI chips into simulating the human brain: A detailed performance analysis. *Neurocomputing*, 598, p.127953. Available from: https://doi.org/10.1016/j.neucom.2024.127953.

[32] Li, A., Sun, J., Wang, B., Duan, L., Li, S., Chen, Y. and Li, H., 2021. LotteryFL: Empower Edge Intelligence with Personalized and Communication-Efficient Federated Learning. *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. pp.68–79. Available from: https://doi.org/10.1145/3453142.3492909.

[33] Li, E., Zeng, L., Zhou, Z. and Chen, X., 2020. Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing. *IEEE Transactions on Wireless Communications*, 19(1), pp.447–457. Available from: https://doi.org/10.1109/TWC.2019.2946140.

[34] Li, X., Bi, S. and Wang, H., 2021. Optimizing Resource Allocation for Joint AI Model Training and Task Inference in Edge Intelligence Systems. *IEEE Wireless Communications Letters*, 10(3), pp.532–536. Available from: https://doi.org/10.1109/LWC.2020.3036852.

[35] Li, X., Wang, S., Zhu, G., Zhou, Z., Huang, K. and Gong, Y., 2022. Data Partition and Rate Control for Learning and Energy Efficient Edge Intelligence. *IEEE Transactions on Wireless Communications*, 21(11), pp.9127–9142. Available from: https://doi.org/10.1109/TWC.2022.3173262.

[36] Li, X., Wang, S., Zhu, G., Zhou, Z., Huang, K. and Gong, Y., 2022. Learning and Energy Efficient Edge Intelligence: Data Partition and Rate Control. *ICC 2022 - IEEE International Conference on Communications*. pp.5353–5358. Available from: https://doi.org/10.1109/ICC45855.2022.9838801.

[37] Li, Y., Zhu, L., Wang, H., Yu, F.R. and Liu, S., 2021. A Cross-Layer Defense Scheme for Edge Intelligence-Enabled CBTC Systems Against MitM Attacks. *IEEE Transactions on Intelligent Transportation Systems*, 22(4), pp.2286–2298. Available from: https://doi.org/10.1109/TITS.2020.3030496.

[38] Lian, Z., Cao, J., Zuo, Y., Liu, W. and Zhu, Z., 2021. AGQFL: Communication-efficient Federated Learning via Automatic Gradient Quantization in Edge Heterogeneous Systems. *2021 IEEE 39th International Conference on Computer Design (ICCD)*. pp.551–558. Available from: https://doi.org/10.1109/ICCD53106.2021.00089.

[39] Liang, T., Glossner, J., Wang, L., Shi, S. and Zhang, X., 2021. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461, pp.370–403. Available from: https://doi.org/10.1016/j.neucom.2021.07.045.

[40] Liao, Y., Xu, Y., Xu, H., Wang, L., Yao, Z. and Qiao, C., 2024. MergeSFL: Split Federated Learning with Feature Merging and Batch Size Regulation. *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. pp.2054–2067. Available from: https://doi.org/10.1109/ICDE60146.2024.00164.

[41] Lin, X., Liu, R., Xie, J., Wei, Q., Zhou, Z., Chen, X., Huang, Z. and Lu, G., 2023. Online Scheduling of CPU-NPU Co-inference for Edge AI Tasks. *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. pp.1–6. Available from: https://doi.org/10.1109/WCNC55385.2023.10118755.

[42] Liu, S., Wen, D., Li, D., Chen, Q., Zhu, G. and Shi, Y., 2024. Energy-Efficient Optimal Mode Selection for Edge AI Inference via Integrated Sensing-Communication-Computation. *IEEE Transactions on Mobile Computing*, 23(12), pp.14248–14262. Available from: https://doi.org/10.1109/TMC.2024.3440581.

[43] Liu, X., Xu, C., Yu, H. and Zeng, P., 2022. Multi-Agent Deep Reinforcement Learning for End-Edge Orchestrated Resource Allocation in Industrial Wireless Networks. *Frontiers of Information Technology & Electronic Engineering*, 23(1), pp.47–60. Available from: https://doi.org/10.1631/FITEE.2100331.

[44] Liu, Y.J., Qin, S., Sun, Y. and Feng, G., 2022. Resource Consumption for Supporting Federated Learning in Wireless Networks. *IEEE Transactions on Wireless Communications*, 21(11), pp.9974–

9989. Available from: https://doi.org/10.1109/TWC.2022.3181611.

[45] Liu, Z., Lan, Q., Kalør, A.E., Popovski, P. and Huang, K., 2023. Over-the-Air View-Pooling for Low-Latency Distributed Sensing. *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. pp.71–75. Available from: https://doi.org/10.1109/SPAWC53906.2023.10304546.

[46] Lu, Y., Huang, X., Zhang, K., Maharjan, S. and Zhang, Y., 2021. Low-Latency Federated Learning and Blockchain for Edge Association in Digital Twin Empowered 6G Networks. *IEEE Transactions on Industrial Informatics*, 17(7), pp.5098–5107. Available from: https://doi.org/10.1109/TII.2020.3017668.

[47] Minh, H.T., Mai, L. and Minh, T.V., 2021. Performance Evaluation of Deep Learning Models on Embedded Platform for Edge AI-Based Real time Traffic Tracking and Detecting Applications. *2021 15th International Conference on Advanced Computing and Applications (ACOMP)*. pp.128–135. Available from: https://doi.org/10.1109/ACOMP53746.2021.00024.

[48] Morafah, M., Chang, H. and Lin, B., 2025. Large Scale Delocalized Federated Learning Over a Huge Diversity of Devices in Emerging Next-Generation Edge Intelligence Environments. *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. New York, NY, USA: Association for Computing Machinery, p.127. Available from: https://doi.org/10.1145/3676536.3697130.

[49] Nan, Z., Han, Y., Yan, J., Zhou, S. and Niu, Z., 2025. Robust Task Offloading and Resource Allocation Under Imperfect Computing Capacity Information in Edge Intelligence Systems. *IEEE Transactions on Mobile Computing*, 24(7), pp.6154–6167. Available from: https://doi.org/10.1109/TMC.2025.3539296.

[50] Nayak, S., Patgiri, R., Waikhom, L. and Ahmed, A., 2024. A review on edge analytics: Issues, challenges, opportunities, promises, future directions, and applications. *Digital Communications and Networks*, 10(3), pp.783–804. Available from: https://doi.org/10.1016/j.dcan.2022.10.016.

[51] Nunez-Yanez, J. and Hosseinabady, M., 2021. Sparse and dense matrix multiplication hardware for heterogeneous multi-precision neural networks. *Array*, 12, p.100101. Available from: https://doi.org/10.1016/j.array.2021.100101.

[52] Ogbogu, C., Joardar, B., Chakrabarty, K., Doppa, J. and Pande, P.P., 2024. Data Pruning-enabled High Performance and Reliable Graph Neural Network Training on ReRAM-based Processing-in-Memory Accelerators. *ACM Transactions on Design Automation of Electronic Systems*, 29(5), p.72. Available from: https://doi.org/10.1145/3656171.

[53] Psaromanolakis, N., Theodorou, V., Laskaratos, D., Kalogeropoulos, I., Vlontzou, M.E., Zarogianni, E. and Samaras, G., 2023. MLOps meets Edge Computing: an Edge Platform with Embedded Intelligence towards 6G Systems. *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. pp.496–501. Available from: https://doi.org/10.1109/EuCNC/6GSummit58263.2023.10188244.

[54] Qiao, D., Li, M., Guo, S., Zhao, J. and Xiao, B., 2024. Resources-Efficient Adaptive Federated Learning for Digital Twin-Enabled IIoT. *IEEE Transactions on Network Science and Engineering*, 11(4), pp.3639–3652. Available from: https://doi.org/10.1109/TNSE.2024.3382206.

[55] Qin, S., Chen, Y., Wang, S., Xie, Z., Wen, M. and Ng, D.W.K., 2024. Integrating Edge Intelligence and Industrial IoT via Learning-Communication Balancing Power Allocation. *ICC 2024 - IEEE International Conference on Communications*. pp.861–866. Available from: https://doi.org/10.1109/ICC51166.2024.10622424.

[56] Qiu, C., Yao, H., Wang, X., Zhang, N., Yu, F.R. and Niyato, D., 2020. AI-Chain: Blockchain Energized Edge Intelligence for Beyond 5G Networks. *IEEE Network*, 34(6), pp.62–69. Available from: https://doi.org/10.1109/MNET.021.1900617.

[57] Qu, S., Li, B., Zhao, S., Zhang, L. and Wang, Y., 2023. A Coordinated Model Pruning and Mapping Framework for RRAM-Based DNN Accelerators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(7), pp.2364–2376. Available from: https://doi.org/10.1109/TCAD.2022.3221906.

[58] Robben, S., Englebienne, G. and Kröse, B., 2017. Delta Features From Ambient Sensor Data are Good Predictors of Change in Functional Health. *IEEE Journal of Biomedical and Health Informatics*, 21(4), pp.986–993. Available from: https://doi.org/10.1109/JBHI.2016.2593980.

[59] Sahu, D., Nidhi, Prakash, S., Pandey, V.K., Yang, T., Rathore, R.S. and Wang, L., 2025. Edge assisted energy optimization for mobile AR applications for enhanced battery life and performance. *Scientific Reports*, 15, p.10034. Available from: https://doi.org/10.1038/s41598-025-93731-w.

[60] Shao, Z., Li, B., Wang, P., Zhang, Y. and Choo, K.K.R., 2025. FedLoRE: Communication-Efficient and Personalized Edge Intelligence Framework via Federated Low-Rank Estimation. *IEEE Transactions on Parallel and Distributed Systems*, 36(5), pp.994–1010. Available from: https://doi.org/10.1109/TPDS.2025.3548444.

[61] Shi, Y., Yang, K., Jiang, T., Zhang, J. and Letaief, K.B., 2020. Communication-Efficient Edge AI: Algorithms and Systems. *IEEE Communications Surveys & Tutorials*, 22(4), pp.2167–2191. Available from: https://doi.org/10.1109/COMST.2020.3007787.

[62] Singh, R. and Gill, S.S., 2023. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, 3, pp.71–92. Available from: https://doi.org/10.1016/j.iotcps.2023.02.004.

[63] Surianarayanan, C., Lawrence, J.J., Chelliah, P.R., Prakash, E. and Hewage, C., 2023. A Survey on Optimization Techniques for Edge Artificial Intelligence (AI). *Sensors*, 23(3), p.1279. Available from: https://doi.org/10.3390/s23031279.

[64] Takeuchi, K., 2023. Neuromorphic Computation-in-Memory System (Invited). *2023 IEEE International Reliability Physics Symposium (IRPS)*. pp.1–4. Available from: https://doi.org/10.1109/IRPS48203.2023.10117704.

[65] Tensorflow lite guide, 2024. Available from: https://www.tensorflow.org/lite/guide.

[66] Tseng, F.H. and Huang, Y.H., 2024. FedBF16-Dynamic: Communication-Efficient Federated Learning with Adaptive Transmission. *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. pp.1–6. Available from: https://doi.org/10.1109/INFOCOMWKSHPS61880.2024.10620860.

[67] Tuli, S., Casale, G. and Jennings, N.R., 2022. SimTune: Bridging the Simulator Reality Gap for Resource Management in Edge-Cloud Computing. *Scientific Reports*, 12, p.19158. Available from: https://doi.org/10.1038/s41598-022-23924-0.

[68] Wang, C., Li, R., Li, W., Qiu, C. and Wang, X., 2021. SimEdgeIntel: A open-source simulation platform for resource management in edge intelligence. *Journal of Systems Architecture*, 115, p.102016. Available from: https://doi.org/10.1016/j.sysarc.2021.102016.

[69] Wang, R., Gao, H., Qiu, H., Luo, L., Chen, M., Dong, Z. and Liu, J., 2025. A Cloud-Edge Intelligence-Based Optimization Method for Distribution Network Partitioning and Operation Considering Simulation Inaccuracy. *IEEE Transactions on Power Systems*, 40(5), pp.3750–3762. Available from: https://doi.org/10.1109/TPWRS.2025.3528889.

[70] Wang, S., Wang, R., Hao, Q., Wu, Y.C. and Poor, H.V., 2020. Learning Centric Power Allocation for Edge Intelligence. *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. pp.1–6. Available from: https://doi.org/10.1109/ICC40277.2020.9148872.

[71] Wang, Z., Zhou, Y., Zou, Y., An, Q., Shi, Y. and Bennis, M., 2023. A Graph Neural Network Learning Approach to Optimize RIS-Assisted Federated Learning. *IEEE Transactions on Wireless Communications*, 22(9), pp.6092–6106. Available from: https://doi.org/10.1109/TWC.2023.3239400.

[72] Wen, H., Wu, Y., Yang, C., Duan, H. and Yu, S., 2020. A Unified Federated Learning Framework for Wireless Communications: towards Privacy, Efficiency, and Security. *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. pp.653–658. Available from: https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162672.

[73] Wu, L., Zhao, C., Wang, J., Yu, X., Chen, S., Li, C., Han, J., Xue, X. and Zeng, X., 2024. A Heuristic and Greedy Weight Remapping Scheme with Hardware Optimization for Irregular Sparse Neural Networks Implemented on CIM Accelerator in Edge AI Applications. *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*. pp.551–556. Available from: https://doi.org/10.1109/ASP-DAC58780.2024.10473919.

[74] Xia, W., Zhang, J., Quek, T.Q.S., Jin, S. and Zhu, H., 2020. Mobile Edge Cloud-Based Industrial Internet of Things: Improving Edge Intelligence With Hierarchical SDN Controllers. *IEEE Vehicular Technology Magazine*, 15(1), pp.36–45. Available from: https://doi.org/10.1109/MVT.2019.2952674.

[75] Xu, L., Sun, H., Zhao, H., Zhang, W., Ning, H. and Guan, H., 2023. Accurate and Efficient Federated-Learning-Based Edge Intelligence for Effective Video Analysis. *IEEE Internet of Things Journal*, 10(14), pp.12169–12177. Available from: https://doi.org/10.1109/JIOT.2023.3241039.

[76] Xu, S., Qian, Y. and Hu, R.Q., 2020. Data-Driven Edge Intelligence for Robust Network Anomaly Detection. *IEEE Transactions on Network Science and Engineering*, 7(3), pp.1481–1492. Available from: https://doi.org/10.1109/TNSE.2019.2936466.

[77] Yang, L., Lu, Y., Cao, J., Huang, J. and Zhang, M., 2021. E-Tree Learning: A Novel Decentralized Model Learning Framework for Edge AI. *IEEE Internet of Things Journal*, 8(14), pp.11290–11304. Available from: https://doi.org/10.1109/JIOT.2021.3052195.

[78] Yang, S. and Chen, B., 2023. SNIB: Improving Spike-Based Machine Learning Using Nonlinear Information Bottleneck. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(12), pp.7852–7863. Available from: https://doi.org/10.1109/TSMC.2023.3300318.

[79] Yining, W., Shujun, H., Xiaodong, X., Rui, M., Haotai, L., Chen, D. and Ping, Z., 2024. Intellicise model transmission for semantic communication in intelligence-native 6G networks. *China Communications*, 21(7), pp.95–112. Available from: https://doi.org/10.23919/JCC.fa.2023-0759.202407.

[80] Yun, S., Choi, W. and Kim, I.M., 2022. Cooperative Inference of DNNs for Delay- and Memory-Constrained Wireless IoT Systems. *IEEE Internet of Things Journal*, 9(17), pp.16113–16127. Available from: https://doi.org/10.1109/JIOT.2022.3152359.

[81] Yun, S., Kang, J.M., Choi, S. and Kim, I.M., 2021. Cooperative Inference of DNNs Over Noisy Wireless Channels. *IEEE Transactions on Vehicular Technology*, 70(8), pp.8298–8303. Available from: https://doi.org/10.1109/TVT.2021.3092179.

[82] Zeng, S., Li, Z., Yu, H., Zhang, Z., Luo, L., Li, B. and Niyato, D., 2023. HFedMS: Heterogeneous Federated Learning With Memorable Data Semantics in Industrial Metaverse. *IEEE Transactions on Cloud Computing*, 11(3), pp.3055–3069. Available from: https://doi.org/10.1109/TCC.2023.3254587.

[83] Zhang, S., Ma, D., Bian, S., Yang, L. and Jiao, X., 2023. On Hyperdimensional Computing-based Federated Learning: A Case Study. *2023 International Joint Conference on Neural Networks (IJCNN)*. pp.1–8. Available from: https://doi.org/10.1109/IJCNN54540.2023.10191707.

[84] Zhang, T., Li, G., Wang, S., Zhu, G., Chen, G. and Wang, R., 2023. ISAC-Accelerated Edge Intelligence: Framework, Optimization, and Analysis. *IEEE Transactions on Green Communications and Networking*, 7(1), pp.455–468. Available from: https://doi.org/10.1109/TGCN.2022.3233913.

[85] Zhang, X., Liu, J., Xiong, Z., Huang, Y., Xie, G. and Zhang, R., 2024. Edge Intelligence Optimization for Large Language Model Inference with Batching and Quantization. *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. pp.1–6. Available from: https://doi.org/10.1109/WCNC57260.2024.10571127.

[86] Zhao, D., Ding, R. and Song, B., 2025. Satellite-assisted 6G wide-area edge intelligence: dynamics-aware task offloading and resource allocation for remote IoT services. *Science China Information Sciences*, 68(2), p.122303. Available from: https://doi.org/10.1007/s11432-024-4258-x.

[87] Zhao, Y., Qu, Y., Xiang, Y., Shi, C., Chen, F. and Gao, L., 2024. Long-Term Over One-Off: Heterogeneity-Oriented Dynamic Verification Assignment for Edge Data Integrity. *IEEE Transactions on Mobile Computing*, 23(5), pp.4601–4616. Available from: https://doi.org/10.1109/TMC.2023.3294180.

[88] Zhao, Y., Xu, C., Qu, Y., Xiang, Y., Chen, F. and Gao, L., 2024. A Learning-Based Hierarchical Edge Data Corruption Detection Framework in Edge Intelligence. *IEEE Internet of Things Journal*, 11(10), pp.18366–18380. Available from: https://doi.org/10.1109/JIOT.2024.3366292.

[89] Zhou, L., Hong, Y., Wang, S., Han, R., Li, D., Wang, R. and Hao, Q., 2021. Learning Centric Wireless Resource Allocation for Edge Computing: Algorithm and Experiment. *IEEE Transactions on Vehicular Technology*, 70(1), pp.1035–1040. Available from: https://doi.org/10.1109/TVT.2020.3047149.