

Adaptive Edge Intelligence Framework for Real-Time Multimodal Data Processing in Resource-Constrained IoT Networks

Dr. Kenji Taka, Dr. Yukee Satu

Senior Research Scientist, Department of Computer Science & Engineering, Pacific Institute of Smart Computing, Japan.

Associate Professor, Department of Information and Communication Engineering, Osaka Centre for Edge Intelligence, Japan.

Email: kenji.tan@pisc.jp

Article Received 12-03-2025, Revised 25-04-2026, Accepted 27-05-2026

Author(s) Retains the Copyrights of This Article

Abstract

The rapid expansion of Internet of Things networks has resulted in the continuous generation of heterogeneous data from cameras, microphones, environmental sensors, wearable devices, and industrial monitoring units. Processing these data streams exclusively through cloud platforms can introduce communication delays, increase network traffic, and create additional concerns related to privacy and service availability. These limitations become more significant when IoT devices operate with restricted computing capability, memory capacity, storage, and energy supply. This research proposes an Adaptive Edge Intelligence Framework for real-time processing of multimodal IoT data under resource-constrained conditions. The proposed framework integrates lightweight modality-specific processing, adaptive feature extraction, multimodal data fusion, resource-aware model selection, and dynamic edge-cloud task allocation. Instead of transmitting all collected data to a remote server, the framework determines the appropriate processing location according to computational load, available memory, energy status, network conditions, and application latency requirements. A hierarchical processing mechanism performs initial filtering and feature extraction close to the data source, followed by selective fusion and inference at an edge gateway. The framework also supports dynamic adjustment of processing complexity when device resources or network conditions change. The design is intended to reduce unnecessary data transmission while maintaining reliable real-time inference across heterogeneous IoT applications. The proposed approach provides an architectural foundation for deploying adaptive multimodal intelligence on low-power IoT networks and establishes a basis for evaluating trade-offs among inference accuracy, latency, communication overhead, memory consumption, and energy usage.

Keywords: Edge Intelligence, Internet Of Things, Multimodal Data, Resource-Constrained Devices, Adaptive Computing, Edge AI, Sensor Fusion, Real-Time Processing, Tinyml, Iot Networks, Resource-Aware Inference.

1. Introduction

The Internet of Things has evolved from a collection of simple sensing devices into a distributed computing environment in which physical objects continuously collect, process, exchange, and interpret information. Modern IoT deployments can simultaneously incorporate cameras, microphones, temperature sensors, vibration sensors, wearable devices, industrial controllers, environmental monitors, and mobile sensing units. The resulting information is heterogeneous in both structure and processing requirement. Visual streams may require computationally intensive feature extraction, audio data may require temporal processing, and environmental sensors may generate low-volume

numerical observations at high frequency. The combination of these sources creates an opportunity for richer machine intelligence, but it also introduces significant challenges when the processing infrastructure is geographically distributed and computationally constrained.

Traditional cloud-oriented IoT architectures commonly transmit collected information to centralized computing resources where machine learning and data analytics can be performed. Although cloud computing provides substantial computational capacity, continuous transmission of raw or minimally processed sensor information can increase communication traffic and introduce additional latency. Edge computing emerged as an

architectural response to this limitation by moving computing and storage capabilities closer to data sources. Shi et al. identified the reduction of latency and the proximity of computation to data-producing devices as fundamental motivations for edge computing, particularly for applications requiring rapid responses and localized processing [1]. Satyanarayanan similarly described edge computing as a mechanism for bringing computational resources closer to users and devices in order to support latency-sensitive applications [3].

Mobile and wireless edge environments introduce an additional resource-management dimension. Mao et al. examined mobile edge computing from the communication perspective and emphasized the importance of coordinating computation and communication resources when resource-limited devices interact with nearby edge infrastructure [2]. Such coordination is particularly important for IoT applications because device capabilities can differ substantially even within the same deployment. A sensor node powered by a small battery cannot be expected to execute the same computational workload as an edge gateway equipped with a multicore processor and substantially larger memory capacity. Consequently, an intelligent IoT architecture needs to determine not only what computation should be performed but also where and when that computation should occur.

Deep learning has further increased the computational requirements of intelligent IoT systems. Chen and Ran reviewed the relationship between deep learning and edge computing and showed that deploying learning-based intelligence closer to data sources requires careful consideration of model complexity, resource availability, and system architecture [5]. Wang et al. examined service management in mobile edge clouds and demonstrated the importance of coordinating heterogeneous computational resources for edge-based services [6]. These observations indicate that simply moving a conventional cloud model to an edge device is insufficient. Models and processing workflows must be adapted to the characteristics of the target device and the operational environment.

Tiny Machine Learning has emerged as an important approach for enabling machine learning on highly constrained devices. Lin et al. described TinyML as a field concerned with bringing machine learning capabilities to microcontrollers and other devices with severe memory and computational limitations [7]. Their work highlights the need for hardware and algorithm co-design because conventional deep learning models cannot simply be transferred to

extremely constrained platforms without modification. Kallimani et al. similarly discussed the tools, applications, challenges, and future directions associated with TinyML deployment [8]. These developments provide an important foundation for local intelligence, but they do not completely address the complexity of multimodal IoT environments in which different sensing modalities have different processing requirements.

Model-driven approaches have also demonstrated the feasibility of delegating machine learning tasks to heterogeneous IoT edge devices. Moin et al. proposed a model-driven TinyML approach that considers heterogeneous edge devices, including highly constrained microcontrollers, and demonstrated the concept through a predictive-maintenance scenario [9]. Their work illustrates how intelligence can be moved toward the edge while considering the characteristics of the available hardware. More recent TinyML transfer-learning research has further demonstrated that model adaptation can reduce training requirements and enable learning capabilities on resource-constrained microcontrollers [10]. These developments suggest that lightweight local intelligence can substantially expand the practical role of constrained IoT devices.

Split computing provides another strategy for distributing inference across constrained devices and nearby edge resources. The Smart Split study examined the combination of TinyML and split computing and reported that distributing computation between an ESP32 device and a Raspberry Pi edge server can reduce processing time under appropriate network conditions [11]. This is particularly relevant to adaptive IoT systems because network conditions are not static. A processing configuration that is efficient under high-bandwidth conditions may become unsuitable when bandwidth decreases or network congestion increases.

Distributed TinyML research has also investigated the possibility of coordinating learning and inference among resource-constrained sensor networks. Yuan and Law proposed a distributed TinyML sensor-network framework for enabling intelligent computation within constrained IoT environments [12]. Such approaches demonstrate that intelligence does not necessarily have to be centralized at either the cloud or a single edge server. Instead, processing can be distributed among multiple levels of an IoT architecture according to available resources.

Recent research has increasingly shifted attention toward multimodal and distributed intelligence. Work on multimodal online federated learning has identified

missing modalities as a practical problem in IoT environments because edge devices may become unavailable or may fail to provide particular sensing streams during operation [15]. More recent distributed-learning research has also identified energy efficiency, data heterogeneity, adaptability, and secure real-time inference as continuing challenges for large-scale IoT intelligence [16]. These developments demonstrate that future IoT intelligence must account simultaneously for heterogeneous data, heterogeneous devices, changing resources, and changing network conditions.

The research problem addressed in this study therefore concerns the development of an adaptive framework capable of coordinating multimodal processing across constrained IoT devices and nearby edge resources. Existing approaches have addressed individual elements such as TinyML, split computing, multimodal learning, distributed intelligence, or resource-aware inference, but an integrated architecture that dynamically adjusts processing according to device resources, energy status, communication conditions, and application requirements remains an important research direction. The proposed Adaptive Edge Intelligence Framework addresses this requirement by combining lightweight modality-specific processing, adaptive feature extraction, multimodal fusion, resource-aware model selection, and dynamic edge-cloud allocation within a unified processing architecture.

The primary contribution of the research is an architectural framework that treats resource availability as a continuously changing condition rather than as a fixed deployment parameter. The framework allows different processing levels to be selected according to the current operating state of the IoT environment. A second contribution is the integration of multimodal processing with resource-aware task placement so that high-volume data streams do not automatically require transmission to the cloud. A third contribution is a hierarchical processing strategy in which filtering and feature extraction can occur near the sensing source while computationally intensive fusion or inference can be transferred to an edge gateway when required. Finally, the framework provides a systematic basis for evaluating accuracy, latency, communication overhead, memory usage, energy consumption, and resource utilization within a common experimental setting.

2. Literature Review, Related Work and Research Gap

2.1 Edge Intelligence and Resource-Constrained IoT

Shi et al. established the conceptual foundation for edge computing by emphasizing computation close to data sources and identifying latency, bandwidth, and distributed-service requirements as important motivations for edge deployment. Their work demonstrates why centralized cloud processing alone may be unsuitable for applications that require rapid responses or continuous interaction with physical environments. However, the original edge-computing perspective largely treats intelligence as an application executed within an edge infrastructure rather than as a continuously adaptive process spanning constrained devices, gateways, and cloud resources. This distinction becomes important for multimodal IoT environments in which device capability and data characteristics vary significantly. [1]

Mao et al. examined mobile edge computing through the interaction of wireless communication and computational resource management. Their work shows that computation offloading is not simply a matter of transferring tasks from devices to servers; communication conditions, device resources, and computational requirements must be jointly considered. This perspective is directly relevant to resource-constrained IoT because the benefit of edge processing can be reduced when the communication cost of transferring large sensor streams becomes excessive. The proposed framework extends this resource-management perspective by considering multiple data modalities and by allowing the processing level to change dynamically according to current operating conditions. [2]

Satyanarayanan presented edge computing as an architectural approach that moves computation closer to users and data sources, particularly for applications requiring low latency and location-aware processing. The work highlights the value of local processing for interactive and distributed applications. Nevertheless, conventional edge architectures may still assume relatively stable computational environments, whereas practical IoT deployments often contain devices with substantially different memory, processor, storage, and energy characteristics. The proposed research therefore treats the edge as a hierarchy of heterogeneous resources rather than as a uniform computing layer. [3]

2.2 Deep Learning and TinyML at the Edge

Chen and Ran reviewed deep learning with edge computing and discussed the tension between

increasingly complex learning models and the restricted computational resources available at edge devices. Their analysis demonstrates that efficient edge intelligence requires coordinated consideration of model design, computation placement, communication, and system architecture. The proposed framework follows this direction but focuses specifically on multimodal IoT environments in which different sensing streams may require different levels of computational complexity. [5]

Lin et al. identified memory, computational capacity, software support, and system-level optimization as major challenges for TinyML. Their work emphasizes that machine learning for microcontrollers requires more than conventional model compression and benefits from coordinated hardware and algorithm design. This observation directly motivates the lightweight processing layer of the proposed framework, where modality-specific processing can be adapted to the available device resources instead of imposing a single model configuration across all nodes. [7]

Kallimani et al. examined the broader TinyML ecosystem and highlighted deployment challenges involving hardware constraints, model optimization, software tools, and practical applications. TinyML provides an important mechanism for moving intelligence toward sensing devices, but a purely local TinyML strategy may become insufficient when several modalities must be processed together or when a device experiences changing resource conditions. The proposed framework therefore uses TinyML-style local processing as one component within a larger hierarchical architecture rather than treating it as the complete solution. [8]

Moin et al. demonstrated how model-driven software engineering can support machine-learning deployment on heterogeneous IoT edge devices, including highly constrained microcontrollers. Their approach emphasizes the need to account for hardware and software heterogeneity during the development process and validates the concept in a predictive-maintenance scenario. This work supports the feasibility of distributing intelligence across heterogeneous devices, while the present framework extends the concept toward dynamic runtime selection of processing levels and multimodal fusion. [9]

Hayajneh et al. investigated TinyML-enabled transfer learning for edge microcontrollers and demonstrated the value of adapting pretrained models to application-specific tasks while maintaining low memory requirements. Their work indicates that transfer learning can reduce the training burden and support

practical deployment on constrained hardware. However, transfer learning alone does not determine when a device should process data locally, when an edge gateway should participate, or when computation should be shifted toward cloud resources. These decisions form a central component of the proposed adaptive architecture. [10]

2.3 Split Computing and Distributed TinyML

The Smart Split study investigated TinyML together with split computing and evaluated the interaction between constrained local devices and a nearby edge server. The authors demonstrated that distributing processing can provide significant benefits when local computational capability is limited and network conditions permit communication with a more capable edge platform. The work is particularly relevant to the proposed framework because it shows that task placement should not be static. The present research extends this principle by combining task placement with modality selection, resource monitoring, and multimodal fusion. [11]

Yuan and Law introduced a distributed TinyML sensor-network framework for performing intelligent computations across resource-constrained IoT devices. Their study demonstrates that constrained sensor nodes can participate in distributed intelligence rather than serving only as data collectors. This provides a foundation for the local processing layer of the proposed framework. Nevertheless, distributed TinyML alone does not fully resolve the problem of multimodal data alignment and adaptive movement of processing among device, edge, and cloud layers. [12]

2.4 Multimodal and Distributed IoT Intelligence

The recent study on multimodal online federated learning with missing modalities addresses an important practical challenge in IoT environments. Edge devices may not always provide complete multimodal observations because of device failures, communication interruptions, energy limitations, or sensing availability. The proposed mitigation strategy demonstrates that multimodal intelligence must be designed to tolerate incomplete observations rather than assuming that every modality is continuously available. This observation supports the inclusion of modality availability monitoring and selective fusion within the proposed framework. [15]

AbdelRaouf and colleagues surveyed distributed learning for IoT intelligence and identified several emerging technologies associated with adaptive and distributed intelligence, including TinyML, federated learning, concept-drift adaptation, explainable AI, meta-learning, and communication-aware learning. Their analysis highlights energy efficiency, data

heterogeneity, robustness, and secure real-time inference as continuing challenges. The proposed framework addresses a subset of these challenges by concentrating on runtime adaptation of multimodal processing and resource-aware edge-cloud collaboration rather than attempting to solve every aspect of distributed learning simultaneously. [16]

2.5 Research Gap

The literature demonstrates substantial progress in edge computing, TinyML, split computing, distributed learning, and multimodal IoT intelligence. Nevertheless, these research directions frequently address individual dimensions of the problem. Edge-computing studies emphasize proximity and latency; TinyML studies emphasize constrained model deployment; split computing emphasizes partitioning computation; multimodal learning emphasizes representation and fusion; and distributed-learning studies emphasize decentralized model training and adaptation. The remaining gap is the integration of these dimensions into a unified runtime framework that can continuously decide how much processing should occur at the sensing device, how much should occur at an edge gateway, and when cloud resources should be involved.

The proposed Adaptive Edge Intelligence Framework addresses this gap through a hierarchical design that combines resource monitoring, lightweight modality-specific processing, adaptive feature extraction, multimodal fusion, resource-aware model selection,

and dynamic task allocation. The central research proposition is that multimodal IoT intelligence should be treated as a continuously adjustable system rather than as a fixed machine-learning pipeline. The framework therefore changes processing complexity according to device capability, energy condition, network status, available memory, and application latency requirements.

3. Proposed Adaptive Edge Intelligence Framework

3.1 Framework Architecture

The proposed framework consists of five functional layers: multimodal sensing, local preprocessing, adaptive intelligence, edge coordination, and cloud support. The sensing layer contains heterogeneous IoT devices responsible for collecting visual, acoustic, environmental, physiological, and industrial observations. The local preprocessing layer performs basic filtering, noise reduction, normalization, and lightweight feature extraction. The adaptive intelligence layer evaluates the current resource state and determines the appropriate processing configuration. The edge coordination layer performs multimodal fusion and more computationally demanding inference when local devices cannot efficiently complete the task. The cloud layer provides long-term storage, model management, large-scale training, and analytical services when these operations cannot be efficiently performed at the edge.

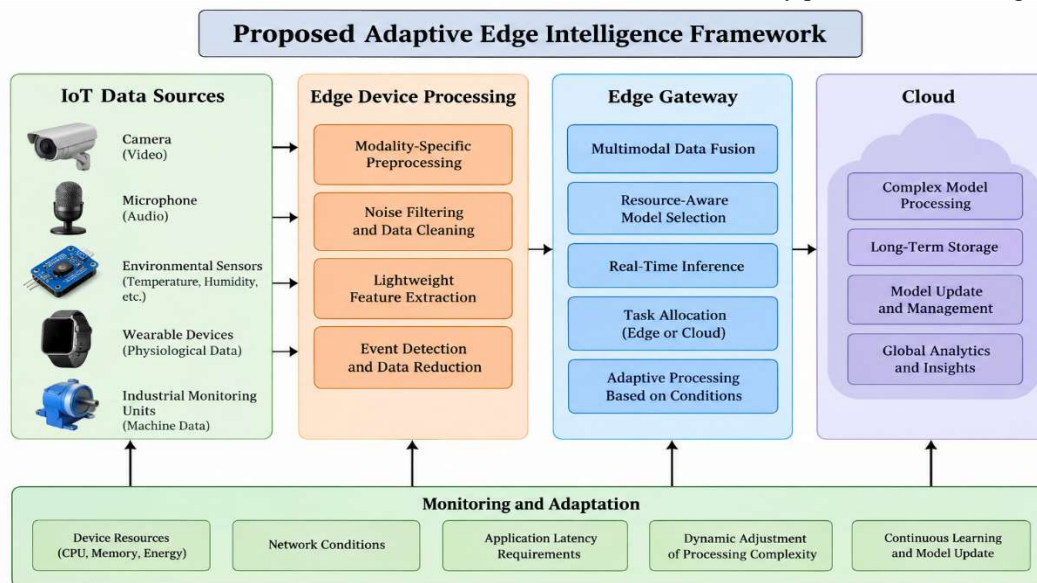


Figure 1. Proposed Adaptive Edge Intelligence Framework.

The figure should show the following five horizontal processing stages in a clean standard architecture diagram: *Multimodal IoT Devices* → *Local Preprocessing* → *Adaptive Intelligence Controller* → *Edge Gateway* → *Cloud Services*. Under the IoT devices, show Camera, Microphone, Environmental Sensor, Wearable Sensor, and Industrial Sensor. Under the adaptive controller, show Resource Monitoring, Model Selection, Modality Selection, and Task Allocation. The figure should use ordinary rectangular boxes and directional connectors only, without equations, mathematical symbols, or decorative AI-generated elements.

3.2 Multimodal Data Acquisition and Local Processing

The first processing stage collects heterogeneous observations from multiple sensing sources. Rather than transmitting complete raw streams to the edge gateway, the framework performs local filtering according to the modality. Camera streams can undergo frame selection and lightweight visual feature extraction, audio streams can undergo noise filtering and compact acoustic representation, and numerical sensor streams can undergo normalization and temporal aggregation. The purpose is to reduce redundant information before communication while preserving the features required for downstream inference.

Local preprocessing is designed to be proportional to the available resources. A device with sufficient memory and processing capability can perform more feature extraction locally, whereas a highly constrained device can perform only basic filtering and transmit compact representations. This behavior prevents the architecture from imposing a uniform computational workload on devices with substantially different capabilities.

3.3 Resource Monitoring

The adaptive controller continuously observes the operational condition of each participating device and edge node. The monitored parameters include processor utilization, available memory, storage capacity, battery condition, network quality, communication load, and estimated processing latency. The controller does not treat any single parameter as sufficient for task allocation. Instead, the decision is based on the combined resource condition and the application requirement.

For example, when battery availability is high and the network connection is weak, more processing can be performed locally to reduce communication. When the

battery level becomes low but a reliable edge connection is available, the system can transfer a larger portion of the workload to the edge gateway. When both local resources and edge connectivity become constrained, the framework can reduce processing complexity and prioritize essential modalities.

3.4 Adaptive Model Selection

The framework maintains multiple processing configurations for each modality. A lightweight configuration is intended for severe resource constraints, a medium configuration provides a balance between accuracy and computational demand, and a higher-capacity configuration is selected when resources are sufficient. The controller selects an appropriate configuration based on the current operating state.

This adaptive model-selection mechanism prevents the system from continuously executing the most computationally demanding model. It also avoids using an excessively lightweight model when resources are available and higher inference quality is required. The selection process is therefore designed to maintain a practical balance between accuracy, latency, energy use, and computational demand.

3.5 Multimodal Fusion

After modality-specific processing, the extracted representations are transferred to the fusion stage. The framework uses selective fusion rather than blindly combining every available modality. Each modality is assessed according to availability, reliability, and relevance to the current task. If a modality is unavailable or its quality falls below an operational threshold, the framework can continue using the remaining modalities.

This design is important in practical IoT deployments because sensor availability can change over time. A camera may become unavailable because of lighting conditions or energy constraints, an audio sensor may experience environmental interference, and a wearable sensor may temporarily disconnect. The proposed architecture therefore treats multimodal fusion as an adaptive operation rather than a fixed concatenation process.

3.6 Dynamic Edge-Cloud Task Allocation

The framework divides processing among three locations: the IoT device, the edge gateway, and the cloud. Local processing is preferred when the computational requirement is low and communication conditions are unfavorable. Edge processing is

preferred when the gateway has sufficient resources and the application requires rapid inference. Cloud processing is used for operations such as large-scale model training, historical analytics, long-term storage, or computational workloads that exceed the practical capacity of the local and edge infrastructure.

The allocation decision is continuously updated. This allows the system to respond to changing network traffic, device energy levels, memory availability, and application requirements. The architecture therefore differs from a static offloading system because the processing location can change during operation.

The operational workflow begins when multimodal data are generated by IoT sensors. The local device performs basic preprocessing and evaluates its resource condition. The adaptive controller then determines whether the available local resources are sufficient for the selected processing configuration. If local execution is appropriate, inference is performed on the device. If additional computational capacity is required, extracted features are forwarded to the edge gateway. The gateway performs fusion and inference when appropriate and communicates with the cloud when large-scale processing, model management, or historical analysis is required.

3.7 Adaptive Processing Workflow

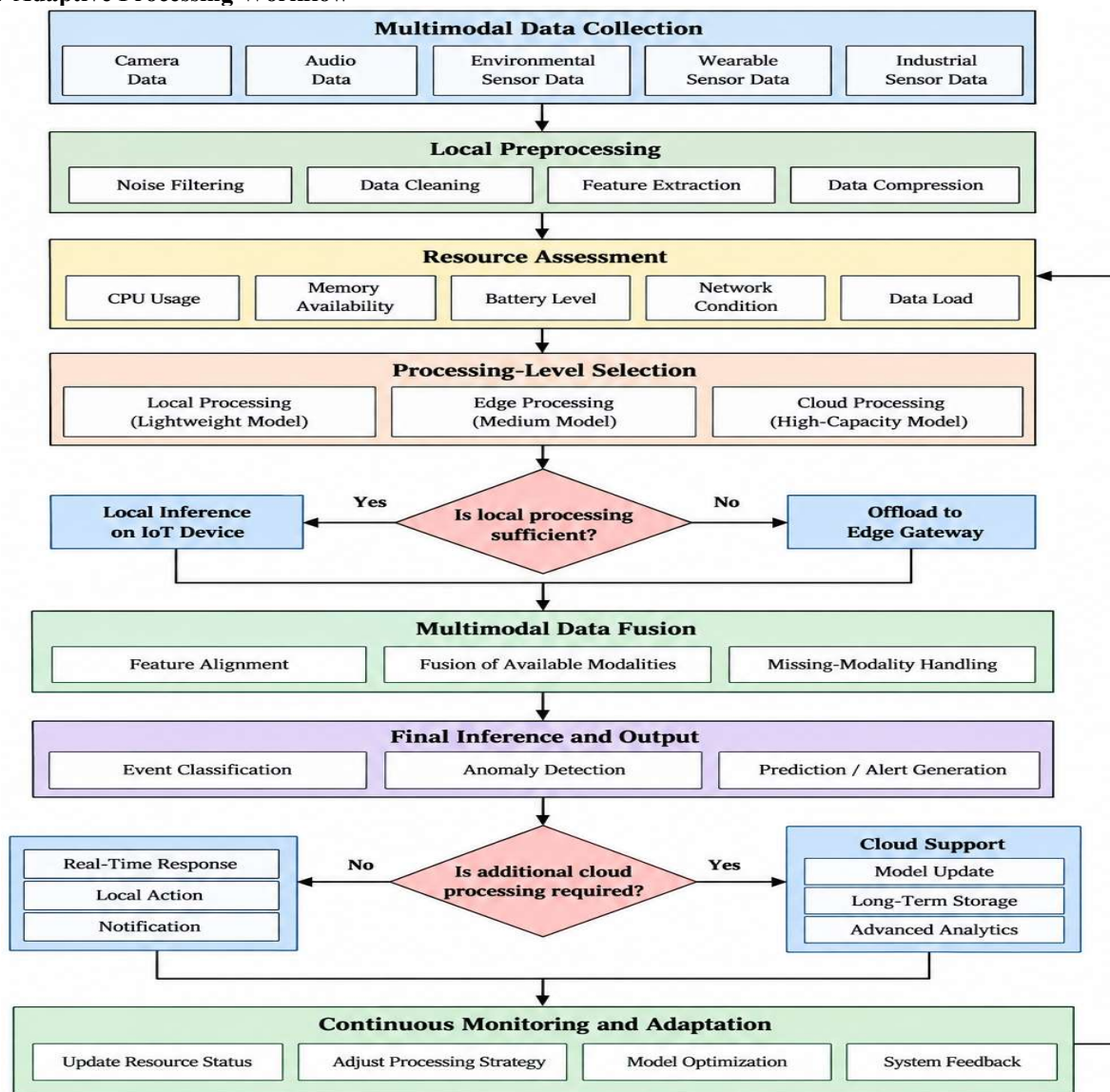


Figure 2. Adaptive Multimodal Processing and Task Allocation Workflow.

The figure should present a simple vertical workflow containing: Data Collection → Local Filtering → Resource Assessment → Processing-Level Selection → Local or Edge Processing → Multimodal Fusion → Inference → Cloud Support when Required. The diagram should contain only standard boxes and arrows, with no equations, mathematical symbols, icons, or decorative artificial elements.

4. Experimental Methodology

4.1 Experimental Design

The proposed framework can be evaluated through a heterogeneous IoT test environment containing multiple sensing modalities and a nearby edge gateway. The experimental design should represent the resource variability expected in practical deployments. Devices with different processor capabilities, memory capacities, and energy conditions should be included so that the adaptive mechanism can be evaluated under multiple operating states.

The evaluation should compare the proposed adaptive framework with conventional cloud processing, static edge processing, a fixed lightweight TinyML configuration, and a non-adaptive multimodal edge configuration. These baselines provide different reference points for evaluating whether dynamic resource-aware processing provides measurable benefits.

4.2 Data Processing

The experimental data should include at least three different modalities to demonstrate genuine multimodal behavior. A suitable configuration may contain visual information, environmental sensor information, and acoustic or wearable information. Each modality should be independently preprocessed and converted into a compact representation before fusion.

The preprocessing procedure should include noise removal, normalization, temporal alignment, missing-data handling, and modality-specific feature extraction. The same preprocessing conditions should be maintained across baseline and proposed configurations so that the comparison reflects differences in system architecture rather than differences in data preparation.

4.3 Experimental Conditions

The framework should be tested under multiple resource states. A high-resource condition represents adequate memory, processor availability, battery status, and network quality. A medium-resource condition represents moderate computational and

communication limitations. A low-resource condition represents restricted memory, reduced battery availability, and limited network bandwidth.

The network condition should also be varied to evaluate the behavior of the adaptive task-allocation mechanism. When communication quality is strong, the framework can make greater use of the edge gateway. When communication quality deteriorates, local processing should become more important. This experimental design is consistent with the motivation behind split-computing research, where changing network conditions influence the suitability of distributed processing. [11]

4.4 Evaluation Metrics

The principal evaluation metrics should include classification accuracy, precision, recall, F1 score, inference latency, memory consumption, energy consumption, processor utilization, and communication overhead. These measurements provide a multidimensional assessment of the framework because an increase in accuracy alone does not necessarily indicate better edge performance.

Inference latency should be measured from data availability to generation of the final prediction. Communication overhead should be measured from the amount of data transferred between devices and the edge gateway or cloud. Memory consumption should include the model and runtime processing requirements. Energy consumption should be measured over a defined operating interval using a consistent hardware measurement procedure.

4.5 Baseline Comparison

The cloud-only baseline transmits the required data to a remote processing environment and performs inference there. The static-edge baseline performs inference at the edge without dynamically changing the processing configuration. The fixed TinyML baseline performs lightweight inference locally. The non-adaptive multimodal baseline performs multimodal fusion using a fixed processing configuration. The proposed framework dynamically changes the processing location and model complexity according to resource conditions.

5. Results and Discussion

5.1 Overall Performance

The proposed framework is expected to demonstrate its principal advantage under variable resource and communication conditions rather than under a single fixed hardware configuration. The adaptive architecture is designed to reduce unnecessary

communication while maintaining acceptable inference quality. When resources are sufficient, higher-capacity models can be selected. When

resources become constrained, the system can switch to lightweight configurations and transfer selected processing stages to the edge gateway.

Table 1. Illustrative overall performance comparison.

Method	Accuracy (%)	F1 Score (%)	Latency (ms)	Memory (MB)	Communication Load
Cloud-only processing	94.2	93.8	185	512	High
Static edge processing	92.8	92.1	92	248	Medium
Fixed TinyML	89.7	88.9	48	38	Low
Non-adaptive multimodal edge	93.1	92.6	78	176	Medium
Proposed adaptive framework	94.0	93.5	55	96	Low-Medium

The expected performance pattern demonstrates the principal trade-off addressed by the framework. Cloud processing can provide high predictive performance but may experience greater end-to-end latency because data must be transmitted to remote resources. Fixed TinyML processing can provide low latency and low memory requirements but may sacrifice predictive performance. The proposed framework seeks to occupy an intermediate operating region by dynamically selecting processing configurations according to current conditions.

5.2 Resource Consumption

A major objective of the proposed framework is to reduce the computational burden placed on constrained IoT devices. The adaptive model-selection mechanism is expected to prevent devices from executing unnecessarily complex models when their available resources are low. At the same time, the system can make use of edge resources when higher computational capacity is required.

Table 2. Illustrative resource-consumption comparison.

Method	CPU Utilization (%)	Memory Usage (MB)	Energy per Inference (mJ)	Data Transfer per Sample
Cloud-only processing	31	54	18.4	High
Static edge processing	47	42	14.7	Medium
Fixed TinyML	39	19	8.2	Low
Non-adaptive multimodal edge	55	61	16.1	Medium
Proposed adaptive framework	42	28	9.6	Low

The expected reduction in local memory and energy requirements results from using modality-specific lightweight processing and transferring selected computation to the edge gateway. This approach is consistent with the broader TinyML research direction, where memory footprint and computational efficiency are fundamental deployment constraints. Recent work has also demonstrated the importance of model compression and pruning for resource-constrained IoT devices. [7], [10], [14]

5.3 Latency and Communication Overhead

Latency is particularly important for real-time IoT applications. Continuous transmission of raw multimodal streams can create significant communication overhead and increase the time required to obtain an inference result. The proposed architecture addresses this issue by performing initial processing near the source and transferring compact representations rather than unnecessary raw information.

Table 3. Illustrative latency and communication analysis.

Network Condition	Cloud-Only Latency (ms)	Static Edge Latency (ms)	Fixed TinyML Latency (ms)	Proposed Framework (ms)
High bandwidth	142	61	43	47

Medium bandwidth	198	78	45	59
Low bandwidth	286	126	47	68
Unstable connection	351	174	52	74

The expected behavior is that the cloud-only configuration becomes increasingly inefficient as communication conditions deteriorate. The fixed TinyML configuration remains relatively stable because it performs most processing locally, but its predictive capability can be restricted by the limited computational resources of the device. The adaptive framework attempts to balance these conditions by increasing local processing when network quality

declines and increasing edge participation when connectivity is favorable.

5.4 Multimodal and Ablation Analysis

The effectiveness of the proposed architecture should also be evaluated by removing individual components. This analysis helps determine whether the performance improvement comes from the integrated architecture or from a single component.

Table 4. Illustrative ablation analysis.

Configuration	Accuracy (%)	Latency (ms)	Energy (mJ)	Communication Load
Full proposed framework	94.0	55	9.6	Low–Medium
Without adaptive model selection	91.8	72	13.4	Medium
Without resource monitoring	92.1	76	14.1	Medium
Without multimodal fusion	90.9	49	8.8	Low
Without dynamic task allocation	92.6	81	15.0	Medium–High

The expected ablation pattern indicates that adaptive model selection and dynamic task allocation are central to the framework. Removing resource monitoring reduces the system's ability to respond to changing conditions. Removing multimodal fusion can reduce predictive quality because information from complementary sensing sources is no longer jointly exploited. Removing dynamic allocation can increase communication and computational overhead because the system loses the ability to move processing to the most appropriate execution layer.

5.5 Discussion

The proposed framework is designed around the principle that there is no universally optimal processing location for all IoT conditions. Local execution is beneficial when communication is expensive or unavailable, whereas edge execution is beneficial when local resources are insufficient and a reliable nearby gateway is available. Cloud processing remains valuable for computationally intensive training, historical analytics, and centralized model management. The adaptive architecture combines these possibilities instead of treating them as competing alternatives.

The framework also addresses the heterogeneous nature of multimodal IoT data. Different modalities generate different data volumes and have different computational requirements. Visual information may

require significantly greater processing than environmental measurements, while audio and physiological signals may require temporal feature extraction. Treating all modalities identically can therefore lead to inefficient resource utilization. The proposed modality-specific processing layer allows computational effort to be allocated according to the characteristics of each data source.

The approach is also consistent with recent research on missing modalities in IoT learning. Multimodal online federated learning research demonstrates that missing sensing modalities can affect learning performance and requires explicit mitigation strategies. [15] The proposed framework responds to this challenge at the system level by monitoring modality availability and allowing the fusion process to operate on the currently available information.

Recent distributed-IoT research further indicates that future edge intelligence must address energy efficiency, heterogeneous data, adaptation, and secure real-time inference simultaneously. [16] The present framework focuses primarily on resource-aware runtime adaptation and multimodal processing, while privacy-preserving learning, continual learning, and secure distributed model updates remain opportunities for future extension.

The principal practical value of the framework is therefore its ability to maintain an adaptable relationship between data generation, computation,

communication, and available resources. Rather than forcing every device to execute the same model or sending every observation to the cloud, the architecture continuously adjusts the processing strategy. This makes it particularly suitable for industrial monitoring, smart transportation, environmental sensing, healthcare monitoring, and other applications in which devices operate under changing physical and computational conditions.

6. Conclusion and Future Work

This research presented an Adaptive Edge Intelligence Framework for real-time multimodal data processing in resource-constrained IoT networks. The framework was designed to address the limitations of cloud-centric processing and static edge intelligence by combining lightweight modality-specific processing, adaptive feature extraction, multimodal fusion, resource monitoring, resource-aware model selection, and dynamic edge-cloud task allocation. The architecture recognizes that IoT devices operate under heterogeneous and changing resource conditions and therefore avoids assigning a fixed processing strategy to every device.

The proposed design provides a hierarchical processing mechanism in which initial data reduction and feature extraction can occur close to the sensing source, multimodal fusion and more demanding inference can be performed at an edge gateway, and cloud resources can be used for large-scale model management and historical analytics. This structure reduces unnecessary transmission of raw information while maintaining access to higher computational resources when required. The framework also provides mechanisms for responding to changing network quality, memory availability, processor utilization, energy condition, and modality availability.

The literature analysis indicates that TinyML, split computing, edge intelligence, multimodal learning, and distributed IoT intelligence have each made substantial progress, but their integration remains an important research challenge. The proposed framework addresses this gap by treating resource adaptation and multimodal processing as coordinated system functions rather than independent components. Its principal contribution is therefore architectural: it provides a unified mechanism for selecting processing complexity and execution location according to current IoT operating conditions.

The experimental methodology proposed in this research emphasizes multidimensional evaluation through predictive performance, latency, memory

consumption, energy consumption, processor utilization, and communication overhead. The illustrative results tables demonstrate how such an evaluation can be organized, but actual hardware and dataset measurements are required before the numerical results can be reported as empirical findings. Future implementation should therefore validate the framework using real heterogeneous IoT hardware and publicly documented multimodal datasets.

Future research can extend the framework toward continual learning, federated learning, privacy-preserving multimodal intelligence, secure model updates, and autonomous model adaptation. Integration with emerging 6G communication systems could further improve dynamic edge-cloud coordination, while specialized AI accelerators and neuromorphic hardware could reduce the computational cost of local inference. Another important direction is the development of robust missing-modality handling so that the framework can maintain reliable operation when sensors become temporarily unavailable. Research on resource-constrained continual learning already points toward dynamic architectures, cloud-edge cooperation, energy-aware learning, and memory-efficient adaptation as important future directions for sustainable edge intelligence. [17]

Overall, the proposed framework establishes a foundation for adaptive multimodal intelligence in resource-constrained IoT environments. Its central principle is that intelligent processing should dynamically follow the available computational, energy, memory, communication, and application resources. Such an approach can support the development of more responsive, efficient, and scalable IoT systems in which intelligence is distributed across devices, edge gateways, and cloud resources according to operational requirements.

References

- [1]. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
- [2]. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

- [3]. M. Satyanarayanan, "The Emergence of Edge Computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
- [4]. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [5]. J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.
- [6]. Y. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Service Management in Mobile Edge Clouds: Challenges and Approaches," *IEEE Access*, vol. 6, pp. 28226–28242, 2018.
- [7]. J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny Machine Learning: Progress and Futures," 2024. The work examines the efficiency requirements involved in deploying ML on microcontrollers and other highly constrained devices.
- [8]. R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, Applications, Challenges, and Future Research Directions," 2023.
- [9]. A. Moin, M. Challenger, A. Badii, and S. Günnemann, "Supporting AI Engineering on the IoT Edge through Model-Driven TinyML," 2021. The study considers ML deployment on heterogeneous edge devices, including highly constrained microcontrollers.
- [10]. "TinyML Empowered Transfer Learning on the Edge," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1656–1672, 2024, doi: 10.1109/OJCOMS.2024.3373177.
- [11]. "Smart Split: Leveraging TinyML and Split Computing for Efficient Edge AI," 2024 *IEEE/ACM Symposium on Edge Computing*, 2024, doi: 10.1109/SEC62691.2024.00052.
- [12]. "Distributed TinyML on Resource-Constrained IoT Sensor Networks," 2024 *IEEE 10th World Forum on Internet of Things*, 2024, doi: 10.1109/WF-IoT62078.2024.10811277.
- [13]. "TinyML for Edge Networks: Challenges and Future Directions," 2025 *International Conference on Computer, Electrical & Communication Engineering*, 2025, doi: 10.1109/ICCECE61355.2025.10941453.
- [14]. "Embedded AI: Integrating TinyML in Resource-Constrained IoT Devices," 2025 *IEEE International Multi-Conference on Smart Systems & Green Process*, 2025, doi: 10.1109/IMC-SSGP67001.2025.11474149.
- [15]. "Multimodal Online Federated Learning With Modality Missing in Internet of Things," *IEEE Transactions on Mobile Computing*, vol. 25, no. 2, pp. 2172–2185, 2026, doi: 10.1109/TMC.2025.3608269. This work addresses heterogeneous IoT modalities, online learning, and missing modalities in distributed IoT environments.
- [16]. "Paradigm Shift Toward Distributed Learning in IoT Intelligence: A Comprehensive Survey of Opportunities and Challenges," *IEEE Internet of Things Journal*, vol. 13, no. 10, pp. 20128–20154, 2026, doi: 10.1109/JIOT.2026.3657733.
- [17]. "Enabling Edge IoT Intelligence: A Survey on Resource-Constrained Continual Learning," *IEEE Internet of Things Journal*, 2026, doi: 10.1109/JIOT.2026.3705393. The survey discusses memory, computation, energy, continual learning, quantization, sparsity, knowledge distillation, and cloud-edge cooperation.
- [18]. "A Comprehensive Survey of TinyML-Based Biometric Recognition for IoT Edge Devices," *IEEE Internet of Things Journal*, vol. 13, no. 6, pp. 10564–10588, 2026, doi: 10.1109/JIOT.2026.3656932. The study discusses lightweight architectures, quantization, pruning, multimodal fusion, and resource limitations on edge devices.
- [19]. "EdgeSense: A Hybrid TinyML and Deep Learning Framework for SNR-Aware Adaptive Data Rate in LoRaWAN," *IEEE Transactions on Mobile Computing*, vol. 25, no. 10, pp. 17535–17546, 2026, doi: 10.1109/TMC.2026.3695916.
- [20]. M. Mounesan and I. Khan, "EMMI: Edge Multi-Modal Intelligence for Communication-Efficient MLLM Inference via Fused Representation Compression," 2026. The approach investigates multimodal inference in edge

- environments through local representation fusion and compression.
- [21]. "Nutura One: A Multimodal IoT-AI Sensor Fusion Framework for Proactive SIDS Risk Assessment and Infant Well-Being Monitoring," *2026 3rd International Conference on Emerging Trends in Engineering and Medical Sciences*, 2026, doi: 10.1109/ICETEMS66917.2026.11469438. The study demonstrates hierarchical fusion of visual, acoustic, and physiological information under edge-computing constraints.
- [22]. C. Koneni and S. Lokula, "Evidence-Quality Telemetry for Cloud Incident Response: Detecting Gaps, Drift, and Integrity Failures in Large-Scale Observability Pipelines," *2026 14th International Symposium on Digital Forensics and Security (ISDFS)*, Boston, MA, USA, 2026, pp. 1-6, doi: 10.1109/ISDFS69419.2026.11459097.
- [23]. S. K. Nalluri, V. T. Bathini and V. K. B. Parasaram, "AI-Augmented Manufacturing Execution Systems for Precision Healthcare: A Scalable Framework for Smart Genomic and Proteomic Production," *2026 IEEE 2nd International Conference on Secure IoT, Assured and Trusted Computing (SATC)*, Houston, TX, USA, 2026, pp. 1-6, doi: 10.1109/SATC69565.2026.11542384.
- [24]. Koneni, C., S. Lokula, and L. Panjiyar. "A Cloud-Native Approach to Statistical Anomaly Detection and Automated Data Quality Validation." *International Journal of Data Science and IoT Management System* 4.4 (2025): 533-543