Volume 4/Issue 10/ October 2026 International Journal of Modern Science and Research Technology
ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

Agentic Edge Intelligence: Integrating Large Language Models, Federated Learning, and Edge Computing for Autonomous Decision Systems

Naseef Talukder
Faculty of ITHM Department, Primeasia University, Dhaka

Abstract: The convergence of large language models (LLMs), federated learning (FL), and edge computing offers a pathway to autonomous decision systems that can reason near data sources while limiting reliance on centralized infrastructure. Edge-deployed LLMs can support latency-sensitive and privacy-conscious applications, although their computational demands challenge resource-constrained devices. FL enables collaborative adaptation without directly pooling private data, but heterogeneous devices, non-identically distributed data, and communication constraints complicate training. This paper proposes an agentic edge-intelligence framework that combines local LLM inference, federated model adaptation, and multi-agent coordination for sensing, planning, and decision execution. The framework identifies interfaces among reasoning, learning, and edge-resource management, and articulates design considerations for privacy, communication efficiency, and device heterogeneity. It provides a systems-oriented basis for developing and evaluating autonomous edge applications under practical network and resource constraints.

Keywords: *Agentic Edge Intelligence; Large Language Models; Federated Learning; Edge Computing; Multi-Agent Systems; Autonomous Decision-Making.*

I. INTRODUCTION

The transition from cloud-centric artificial intelligence toward connected intelligence places greater emphasis on processing data and making decisions near its source, where edge computing can reduce latency and support privacy-sensitive applications. Yet deploying large language models (LLMs) at the edge remains constrained by device resources, while centralized model training raises concerns about data privacy and communication costs. Federated learning (FL) offers a means of collaborative model training without directly collecting local data, but heterogeneous devices and networks complicate efficient deployment.

Integrating LLMs with FL and edge computing creates the possibility of autonomous systems that combine local inference with distributed learning, but these components must operate under resource, communication, and privacy constraints. A further challenge is enabling edge analytics to adapt as data patterns and user questions change, rather than relying on hard-coded processes that require redeployment. This document examines the design space for agentic edge intelligence, focusing on how LLM-based reasoning, federated adaptation, and edge execution can be integrated to support autonomous decision systems.

Connected Intelligence, Low-Latency Decision-Making, and the Case for Edge Deployment

Connected intelligence links sensors, devices, vehicles, infrastructure, and human users through distributed computation and communication. Cloud-centric AI concentrates data processing and model execution in remote data centres, which can impose delay, bandwidth demand, and exposure of sensitive data. **Edge intelligence** relocates inference and parts of learning toward data sources, supporting time-sensitive applications in vehicles, healthcare, industrial control, drones, and mobile networks [1], [2]. Large language models extend this architecture from fixed analytics toward systems that interpret goals, generate plans, write task-specific code, and coordinate heterogeneous models [2].

The resulting concept, **agentic edge intelligence**, combines local perception, language-mediated reasoning, federated adaptation, and constrained actuation. Its value depends on more than model accuracy. An autonomous system must respond within operational deadlines, preserve private information, tolerate unstable wireless links, and adapt to changing data and device capacity. Surveys of mobile edge intelligence therefore treat model compression, caching, training, and inference as coupled deployment problems rather than isolated algorithmic tasks [3], [4].

Corpus Scope and Evidence Map Across LLM, Federated Learning, and Edge Systems Research

The evidence considered here spans five connected strands: edge deployment surveys; federated optimization and heterogeneous learning; federated LLM fine-tuning; collaborative inference and model partitioning; and LLM-assisted or multi-agent autonomy. Primary studies report system measurements such as latency, throughput, communication cost, convergence, accuracy, energy, and resilience, whereas surveys organize unresolved design problems [5], [6]. The corpus also includes application-oriented work in autonomous driving, IoMT, industrial systems, UAV networks, and satellite learning [7], [8]. Because many agentic edge

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

proposals remain simulations or frameworks, claims about autonomous decision quality require more caution than claims about communication or resource optimization.[9]

Architectures for LLM-Enabled Edge Intelligence

Device–Edge–Cloud and Multi-Tier Architectures for Distributed Intelligence

An edge LLM system distributes functions across **devices**, **edge servers**, and **cloud services**. Devices collect multimodal data, execute compact models, and apply immediate control policies. Edge servers aggregate local information, host larger models, coordinate clients, and cache frequently requested knowledge. The cloud retains broad models, long-horizon training, global orchestration, and code-generation capacity when local resources are insufficient [3], [10]. This division is a response to unequal compute, memory, energy, and connectivity rather than a fixed three-layer prescription.[11]

A hierarchical design can aggregate updates or representations before they reach the cloud. EdgeFed moves part of local updating to edge servers, reducing mobile computation and global communication relative to FedAvg across bandwidth conditions [12]. Hierarchical federated learning similarly uses cluster heads for intermediate aggregation, reducing dependence on a single controller while introducing resource-allocation and incentive problems [13]. Topology-aware FL surveys distinguish star, hierarchical, and other network arrangements because physical connectivity, mobility, and failure patterns affect both delay and convergence [6].

The appropriate placement depends on task urgency and information sensitivity. Reflexive safety actions should remain local; contextual planning can use an edge model; broad synthesis or model revision can invoke the cloud. LEI illustrates a related pattern in which a cloud-hosted LLM generates candidate lightweight programs, tests them against device data and constraints, and deploys tailored logic to heterogeneous devices [10]. This arrangement expands adaptability while making verification of generated programs a system requirement.

On-Device Inference, Collaborative Edge Execution, and LLM Model Placement

On-device inference minimizes round-trip delay and limits transmission of raw observations, but Transformer models impose memory, arithmetic, and energy demands that many endpoints cannot sustain in full form [14]. Quantization, pruning, compact architectures, and parameter-efficient adaptation reduce these demands, although compression can alter reasoning quality and generation speed [4]. The design question is therefore where each model component should execute, not whether all intelligence belongs on the device.

Collaborative execution assigns portions of a model to several devices or to a device-edge path. EdgeShard partitions an LLM into deployable shards and jointly selects devices and partition points under heterogeneous bandwidth and model complexity. On a real-world Llama2 testbed, it reported up to 50% lower latency and a twofold throughput improvement over the compared method, without accuracy loss [15]. These results concern inference, not federated training, and should not be generalized to all wireless conditions.

Placement can also be adaptive. A system may select a compact local model when the link is congested, offload selected layers when edge capacity is available, or query the cloud for tasks requiring broader context. Mobile edge intelligence surveys identify caching and delivery as complements to inference, since repeated model transfers can otherwise dominate the cost of distributed service [3]. Placement policies must therefore observe queueing, channel quality, battery state, model size, and task deadline.[16]

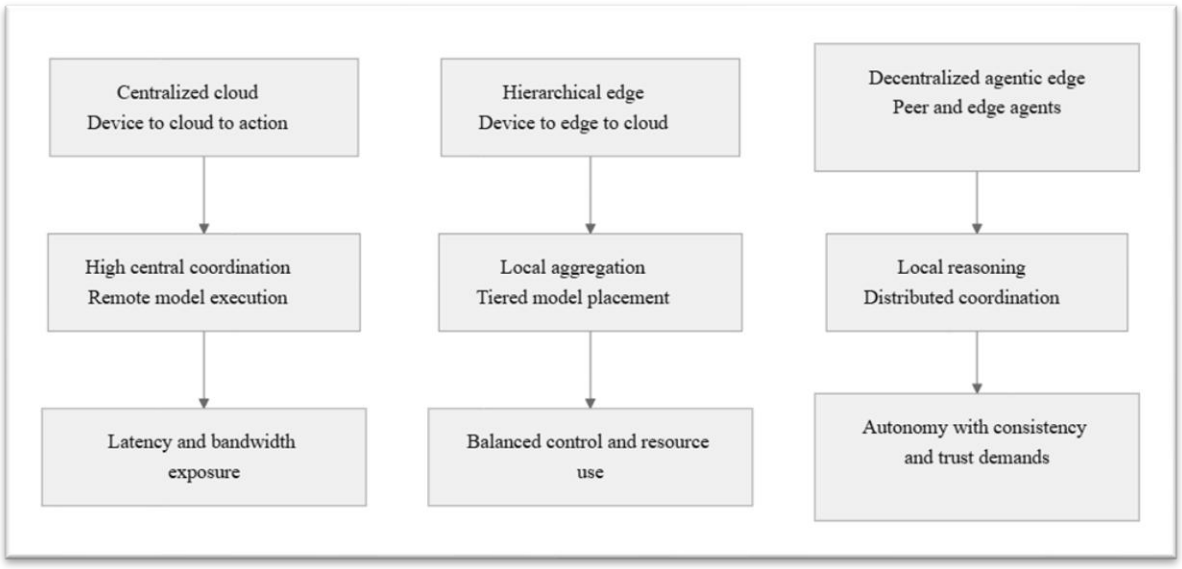
Collaborative Edge Inference Through Sharding, Splitting, and Model Parallelism

Model parallelism divides computation across workers, whereas data parallelism gives each client a complete model and distributes training examples. Split federated learning combines these arrangements by placing early layers on clients and later layers on servers, reducing local memory requirements while transmitting intermediate activations [17]. FedsLLM combines split learning with LoRA and federated aggregation, then optimizes computation and communication jointly; its reported optimization reduced training delay by an average of 47.63% relative to an unoptimized configuration [18]. SflLLM similarly optimizes subchannel allocation, power control, split position, and LoRA rank for heterogeneous wireless clients [19].

The benefit is conditional. Activations and gradients remain communication objects, split points affect privacy exposure, and sequential dependencies can create pipeline bottlenecks. Adaptive local updating and batch-size control in AdaSFL reduced completion time by about 43% and network traffic by about 31% at similar test accuracy in its experiments [17]. These gains reflect a training configuration rather than an inherent property of splitting. EdgeShard addresses inference through shard placement, whereas SflLLM and FedsLLM address distributed fine-tuning; conflating these endpoints obscures their different latency and privacy mechanisms.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

Architectural Comparison Diagram: Centralized Cloud, Hierarchical Edge, and Decentralized Agentic Systems
The three architectures differ in control location, communication path, and autonomy. Centralized cloud systems simplify coordination but transmit observations or requests over longer paths. Hierarchical systems place aggregation and model services at intermediate nodes. Decentralized agentic systems allow peer agents to exchange knowledge and coordinate actions, but they require mechanisms for consistency, trust, and conflict resolution [6], [20].



A practical system may combine all three rather than select one exclusively. Devices can execute safety-critical actions, edge clusters can coordinate nearby agents, and cloud services can update global policies. This compositional view motivates federated learning as the training and personalization substrate.

Federated Learning as the Training and Personalization Substrate for Edge LLMs
Federated Fine-Tuning, Parameter-Efficient Adaptation, and Split Learning for Large Language Models
Federated learning trains a shared model while retaining raw data on participating clients. For LLMs, full-parameter exchange is usually impractical because model size multiplies communication, memory, and local optimization costs. Federated parameter-efficient fine-tuning instead updates smaller components such as LoRA adapters, while preserving most pretrained parameters. FederatedScope-LLM supplies reproducible pipelines and PEFT implementations for instruction tasks, resource reduction, and varied protection requirements [21]. OpenFedLLM covers federated instruction tuning and value alignment across eight training datasets and more than 30 evaluation metrics; its experiments found FL algorithms outperforming local training across the reported settings [22].

The apparent privacy advantage of retaining data locally does not eliminate exposure. Model updates, gradients, activations, and adapter parameters may encode information about local examples. Split learning changes the exposure surface by keeping some layers on the client, but transmitted intermediate states remain subject to inference and interception risks. FedsLLM combines LoRA, split learning, and federated aggregation to reduce client processing, while SflLLM jointly selects split positions and LoRA ranks according to communication and computation conditions [18], [23].

Personalization is necessary when clients have different vocabularies, tasks, preferences, or modalities. A single global LLM can provide general capabilities, while local adapters or task-specific subnetworks capture context that should not be shared. This arrangement creates a tension between cross-client transfer and local specialization. FedLLM-Bench addresses part of the evaluation problem with four datasets, eight training methods, six evaluation metrics, and client populations ranging from 38 to 747, including multilingual, quality, quantity, instruction, length, embedding, and preference variation [24]. Such diversity is more informative than artificially balanced partitions, but benchmark coverage remains narrower than operational edge deployments.[25]

Knowledge Distillation, Dataset Distillation, and Knowledge Caching for Resource-Constrained Clients
Knowledge distillation transfers outputs, logits, representations, or rationales from a teacher to a smaller or heterogeneous student. It avoids requiring every client to hold an identical architecture, a constraint that is particularly restrictive for LLM-enabled edge systems. Fed-ET trains heterogeneous small models on clients and uses weighted consensus distillation with diversity regularization to train a larger server model, reducing communicated parameters while addressing data heterogeneity [26]. KD-based FL can therefore support clients whose model capacity differs, although the server must aggregate knowledge without erasing rare local information.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

Dataset distillation compresses training data into compact synthetic or representative sets, while model distillation compresses capability into a smaller model. The LLM distillation survey identifies task-specific alignment, rationale-based training, multi-teacher methods, gradient matching, latent regularization, and generative synthesis as major directions, while warning that emergent reasoning and linguistic diversity may be lost [27]. These methods are complementary: a small distilled dataset can reduce local training cost, and a distilled student can reduce inference cost. Neither guarantees preservation of long-context planning or safety behaviour.

Knowledge caching reduces repeated parameter exchange by storing reusable information at the server or edge. FedCache uses logits interaction rather than full model-parameter interaction, matching device samples to server-side cached knowledge and applying ensemble distillation locally. Across four datasets, it achieved comparable performance to state-of-the-art personalized FL methods while reporting more than two orders of magnitude improvement in communication efficiency [28]. FedLMA extends this idea through local soft-label extraction, LLM-based assessment of class difficulty and knowledge retention, and selective knowledge filtering in a decentralized IoMT setting [29]. The method addresses client drift, but cached knowledge introduces freshness, provenance, and poisoning questions.[30]

Communication-Efficient Aggregation Through Over-the-Air Computation, Scheduling, and Compression

Communication is often the dominant cost in edge FL because clients repeatedly upload updates over constrained wireless links. **Over-the-air computation** uses waveform superposition to aggregate updates in one wireless transmission, replacing sequential digital uploads with analog aggregation. The approach requires joint device selection and beamforming, formulated in one study as a sparse and low-rank optimization problem [31]. Channel fading and noise create aggregation errors, so power control must account for their effect on convergence rather than merely minimize per-round transmission error [32].

Scheduling determines which clients participate and when. Optimal probabilistic scheduling can minimize total communication time by balancing the number of rounds against per-round latency; the reported policy shifts from reducing remaining rounds early in training toward reducing per-round delay later [33]. Edge-based clustering and cosine-similarity filtering reduced uploaded local updates by 60% and increased convergence speed by 10.3% relative to traditional FL in the reported experiments [34]. These gains depend on the assumption that similar or oppositely directed updates can be identified without harming useful diversity.

Compression operates at several levels: gradients, model parameters, activations, logits, and learned adapters. PruneFL adaptively prunes parameters during federated training and reports similar final accuracy with reduced training time on edge devices including Raspberry Pi [35]. FedDrop assigns heterogeneous subnets according to channel state, reducing communication and computation relative to conventional FL [36]. Compression can conflict with privacy, since sparse or distinctive updates may remain identifiable, and with fairness, since weak clients may repeatedly receive smaller models.[37]

Heterogeneity in Data, Models, Devices, Networks, and Energy Budgets

Edge clients differ across at least five dimensions: statistical data distribution, model architecture, computation capacity, communication conditions, and energy availability. The HFL survey organizes these as statistical, model, communication, and device heterogeneity, with additional challenges arising from their interaction [5]. Non-IID data can cause client drift, slow convergence, and poor global performance. FlocOff uses computation-offloading relationships to reshape participation and reports 14.3%-32.7% improvements in convergence and accuracy under its tested distributions [38].

Synchronous aggregation waits for slow clients, whereas asynchronous aggregation risks stale or imbalanced updates. HHFEL addresses this tension with semi-asynchronous device-to-edge aggregation and elastic cloud-edge updates, while treating device selection and resource allocation as an online decision problem [39]. Energy adds a longer-term constraint: selecting the fastest devices may exhaust them and reduce future participation. FEPCS therefore combines residual energy, transmission estimates, participation progression, energy-budget deviation, and data contribution in client selection, reporting higher accuracy and fewer dropouts under equal energy constraints [40].

Topology and mobility further alter the feasible design. Satellite FL must contend with short ground-to-satellite contact periods and intermittent links; FEDMEGA aggregates within orbits through inter-satellite links and reports approximately 30% faster convergence than compared satellite FEEL methods [8]. A federated LLM system must thus treat architecture, optimization, scheduling, and model format as co-designed variables.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

Comparative Evidence Table: Federated LLM Training Strategies and Their Targeted Constraints

Strategy	Main mechanism	Targeted constraint	Reported evidence
Federated PEFT	LoRA or other adapter updates	LLM memory and communication	FederatedScope-LLM reports resource-efficient PEFT pipelines [21]
Split federated LLM training	Client-server model partitioning with LoRA	Client computation and memory	FedsLLM reports 47.63% lower delay after communication-computation optimization [18]
Knowledge distillation	Logits, soft labels, or ensemble transfer	Model heterogeneity and local capacity	Fed-ET reduces communicated parameters and remains robust to heterogeneous data [26]
Knowledge caching	Cached personalized logits and ensemble distillation	Repeated parameter exchange	FedCache reports more than two orders of magnitude communication improvement [28]
Over-the-air aggregation	Wireless superposition with beamforming	Uplink bandwidth and aggregation time	Device selection and beamforming are optimized jointly [31]
Adaptive pruning and dropout	Dynamic sparsity or heterogeneous subnets	Computation, bandwidth, and energy	PruneFL preserves similar accuracy; FedDrop adapts subnet size to channels [35], [36]

These strategies address different bottlenecks and are therefore composable, but their costs accumulate. The next question is how the resulting models participate in reasoning and action rather than only in prediction.

Agentic Reasoning and Multi-Agent Coordination at the Edge
From Static Edge Analytics to LLM-Assisted Autonomous Planning and Action
Traditional edge analytics maps sensor inputs to predefined predictions or control outputs. **Agentic intelligence** adds a perception-reasoning-action loop in which a system interprets an objective, selects tools or models, plans intermediate steps, observes outcomes, and revises its action. LLMs contribute language understanding, task decomposition, planning, and code generation, while compact local models provide rapid perception and control [2]. The distinction is functional: an agent can change the procedure used to solve a task, whereas static analytics usually executes a fixed pipeline.[41]

LEI addresses changing edge requirements by having a cloud LLM generate candidate device-side programs, test them against data and resource constraints, and deploy a selected program to each device. Across air-quality, temperature and humidity, wind, and soil datasets, the framework maintained low average CPU and memory utilization during execution [10]. The architecture shifts human effort from manually writing every rule toward supervising generated logic. It also creates failure modes involving incorrect code, stale context, unsafe actions, and inadequate validation.

A safe agentic design should separate deliberative planning from time-critical actuation. Local controllers can reject actions that violate physical or operational constraints, while edge or cloud agents propose higher-level plans. This separation is consistent with the resource asymmetry identified in edge LLM surveys: large models may remain off-device even when small models execute locally [3]. Autonomy therefore means conditional delegation and adaptation, not unrestricted model control.

Multi-Agent Reasoning, Knowledge-Guided Decision-Making, and Cooperative Edge Behavior
Multi-agent reasoning assigns complementary roles to distributed agents and combines their observations, plans, or critiques. KoMA uses interaction, multi-step planning, shared memory, and ranking-based reflection for autonomous driving. Its agents infer surrounding-vehicle intentions from scene descriptions, refine actions over several planning steps, retain collective experience, and evaluate candidate behaviour [7]. These modules address reasoning consistency and generalization, but text-based intermediate representations may omit information needed for hard real-time control.

FedLMA shows a second form of cooperation in which LLM-driven agents evaluate class difficulty and knowledge retention, then construct demand vectors for selecting relevant knowledge fragments. The student update combines cross-entropy and KL-divergence losses under highly non-IID IoMT data [29]. Here, agents coordinate learning rather than physical action. The separation matters because a better federated classifier does not by itself establish safe autonomous behaviour.[42]

Multi-agent edge systems also require communication protocols, shared state, and trust rules. A proposed 6G architecture assigns agents to device, edge, and cloud layers and uses hierarchical multi-agent reinforcement learning for orchestration. Its reported reliability approached 99.8%, latency settled around 0.48-0.55 seconds, and energy remained approximately 0.0175-0.0195 joules per task in the described evaluation [43]. These measurements concern orchestration performance in that study, not general LLM reasoning quality. Decentralized designs must additionally manage contradictory beliefs, stale knowledge, malicious agents, and partitioned networks.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

Applications in Autonomous Driving, Industrial Systems, IoMT, Satellite Networks, and Connected Intelligence

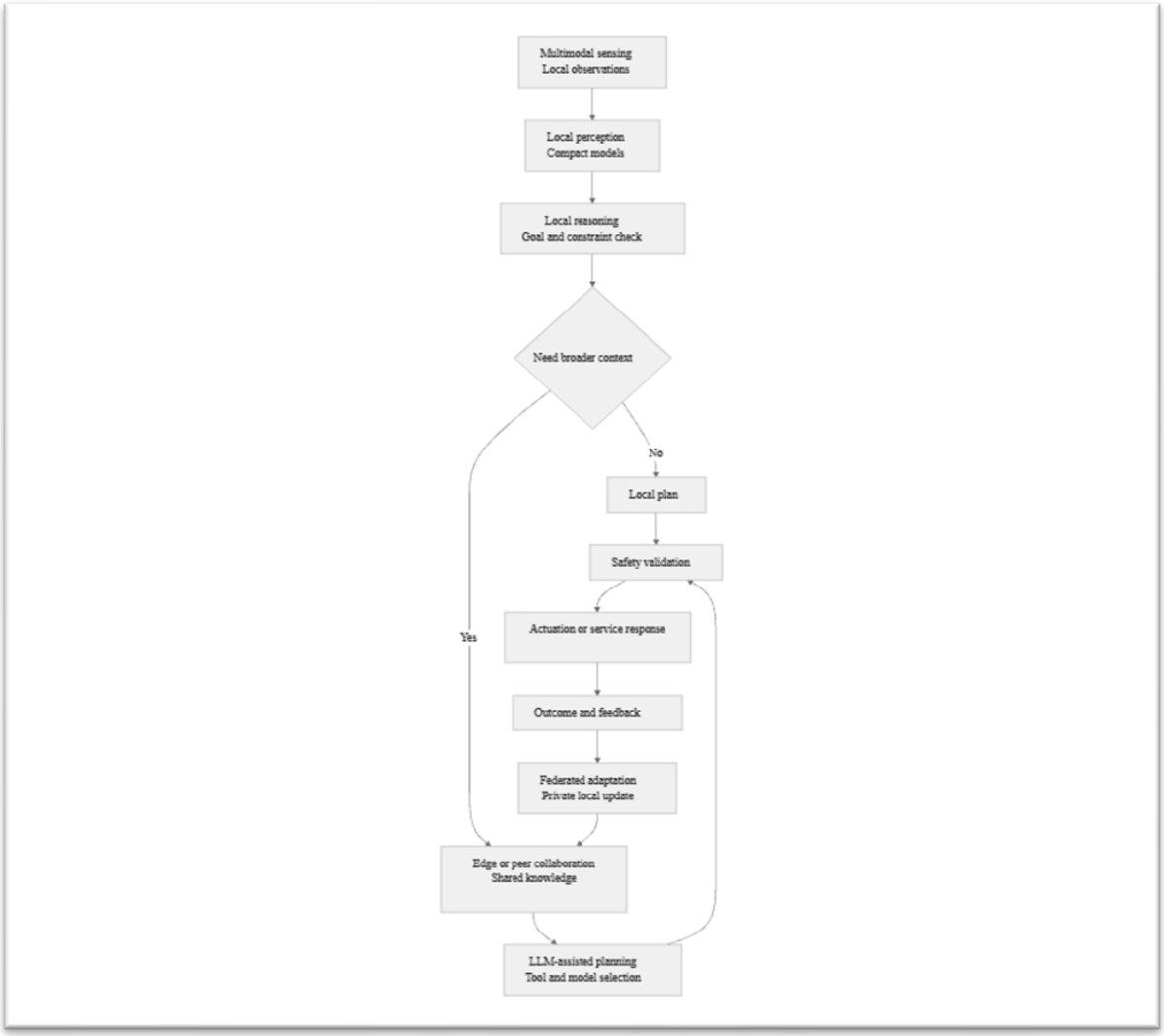
Autonomous driving illustrates why perception, prediction, planning, and coordination cannot be optimized independently. A vehicle agent may use local sensing for immediate collision avoidance, an edge cluster for shared traffic context, and a federated process for learning from distributed driving data. KoMA reports improved handling of complex and unpredictable driving scenarios through cooperative reasoning, shared memory, and reflection [7]. Federated edge learning for collaborative three-dimensional object detection has also been used to evaluate communication-aware scheduling policies [33].

Industrial systems require low delay and protection of operational data. Hybrid differential privacy and adaptive compression have been proposed for industrial federated edge learning to reduce gradient transmission and inference-attack exposure [44]. Industrial agents can combine sensor diagnosis, maintenance planning, and local actuation, but the evidence in the corpus does not establish end-to-end safety under production conditions. IoMT faces similar constraints with more direct privacy implications; FedLMA targets biased local distributions, resource limits, and decentralized knowledge sharing in consumer health scenarios [29].

Satellite networks add intermittent connectivity and mobility. FEDMEGA uses intra-orbit aggregation through inter-satellite links to reduce reliance on lower-rate ground links and reports approximately 30% faster convergence [8]. UAV and connected-intelligence systems require rapid adaptation to changing topology and mission objectives; multi-agent LLM architectures have been proposed for UAV solution generation in mobile edge networks [45]. Across these domains, the strongest evidence concerns architecture and resource handling, while claims about autonomous mission success remain comparatively under-tested.

Agentic Edge Intelligence Workflow Diagram: Perception, Local Reasoning, Collaboration, and Actuation

The workflow below separates the recurrent functions that appear across the agentic edge studies. Local perception and reflexive action limit delay; collaboration supplies broader context; federated updates improve future models without centralizing raw data [2], [7].



ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

The feedback path distinguishes operational adaptation from model training. An agent may revise a plan immediately, whereas federated aggregation changes a shared model over longer rounds. This temporal separation is necessary for evaluating both responsiveness and learning.

System-Level Trade-Offs: Latency, Accuracy, Energy, Privacy, Robustness, and Scalability

Latency and Bandwidth Benefits of Local Inference and Edge Offloading

Local inference removes or shortens the request path to a remote cloud and reduces transmission of raw sensor streams. EdgeShard reports up to 50% latency reduction and twofold throughput improvement through collaborative inference, while EdgeFed reduces global communication and client computation by moving part of updating to edge servers [12], [15]. These results support edge placement for delay-sensitive tasks, but latency depends on model size, queueing, wireless conditions, and synchronization.[46]

Offloading is not uniformly beneficial. Sending activations, gradients, prompts, or model fragments can shift rather than remove communication cost. Split learning may reduce client memory while creating repeated forward and backward transfers. Over-the-air aggregation reduces uplink exchange through wireless superposition, but channel noise and fading can introduce learning error [32]. Thus, a meaningful latency evaluation should report inference time, communication time, waiting time, and end-to-end response time separately.

Energy-Aware Client Selection, Resource Allocation, and Computational Efficiency

Training consumes energy through local optimization, radio transmission, reception, and repeated participation. Cost-effective FL studies link the number of selected clients and local iterations to total learning cost and convergence, showing that these controls should be adapted jointly rather than fixed across rounds [47]. FEPCS adds residual energy and client survival to selection, reducing dropout under a common energy constraint while maintaining higher test accuracy in its simulations [40]. These findings challenge policies that select clients only by speed or instantaneous channel quality.

Compression and partial models reduce arithmetic and communication. PruneFL adapts model size during training while maintaining accuracy similar to the original model on tested edge devices [35]. FedDrop assigns channel-dependent subnets, and module-based federated learning assigns different subnetworks according to local resources [36], [48]. Such methods improve feasibility but can produce unequal learning opportunities and make global model behaviour harder to characterize.

Privacy Preservation Versus Model-Update Leakage and Security Exposure

FL retains raw data locally, but model updates are not automatically private. Parameters, logits, gradients, and activations can reveal information about local examples, while a malicious server or client may manipulate training. Differential privacy reduces inference risk by adding noise, yet utility can decline sharply. In one bandwidth-constrained study, differential privacy applied to KD-based FL caused up to 70% accuracy loss for privacy budget $\varepsilon \leq 10$ [49]. This result demonstrates that privacy protection must be evaluated jointly with the exact task and model.

Secure aggregation, cryptographic protocols, and split learning reduce some exposure but add computation or communication. A privacy-preserving Byzantine-robust scheme using three-party computation increased error by only 1.05% at 50% Byzantine participants, compared with 23.78% without protection in the reported experiments [50]. ROFED-LLM combines split FL, differential privacy, dynamic pruning, adaptive beamforming, and resource allocation; it reports a 12.87% improvement in privacy preservation and an 18.26% improvement in jamming resilience relative to selected baselines, with a 3.94% accuracy trade-off [51]. These values are framework-specific and do not establish a universal privacy-accuracy curve.[52]

Robustness Against Byzantine Behavior, Adversarial Wireless Conditions, and Non-IID Data

Robustness has at least three layers: statistical robustness to non-IID data, systems robustness to dropout and channel variation, and security robustness to malicious updates or interference. Client drift can slow convergence when local distributions differ, motivating personalized models, distillation, topology-aware aggregation, and data-aware scheduling [5], [53]. FedLMA filters knowledge according to local learning difficulty and retention, while FlocOff uses offloading relationships to reduce distributional divergence [29], [38].

Wireless attacks create a separate failure mode. Jamming can prevent timely updates even when the learning algorithm is statistically sound. ROFED-LLM addresses this through adaptive beamforming and model-level protections, combining communication and optimization defences [51]. Satellite systems face non-malicious disruption from mobility and short contact windows, which FEDMEGA addresses through intra-orbit aggregation [8]. Robustness claims should therefore specify whether the endpoint is accuracy, convergence, task completion, link availability, or survival of participating clients.

Evaluation Dashboard: Metrics, Reported Improvements, Resource Costs, and Deployment Constraints

A credible evaluation should report accuracy or task quality alongside latency, throughput, communication volume, energy, memory, client participation, privacy loss, robustness, and safety violations. The studies below illustrate the range of endpoints and why they cannot be collapsed into one performance score.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

System or method	Reported endpoint	Reported result	Main cost or limitation
EdgeShard	Inference latency and throughput	Up to 50% lower latency; 2× throughput	Requires collaborative shard placement and suitable links [15]
AdaSFL	Training completion and network traffic	About 43% lower completion time; about 31% lower traffic	Applies to adaptive split federated training [17]
FedsLLM	Training delay	Average 47.63% reduction after optimization	Requires split placement, bandwidth allocation, and LoRA configuration [18]
FedCache	Communication efficiency	More than two orders of magnitude improvement	Depends on cached knowledge and sample similarity [28]
ROFED-LLM	Privacy and jamming resilience	12.87% and 18.26% improvements; 3.94% accuracy trade-off	Combines DP, pruning, beamforming, and resource optimization [51]
P2Brofl	Byzantine robustness	Error increase of 1.05% at 50% Byzantine clients	Uses secure multi-party computation and protected selection [50]
FEDMEGA	Satellite FL convergence	Approximately 30% improvement	Depends on inter-satellite links and orbital aggregation [8]

The dashboard indicates that edge intelligence has no single optimum. Lower latency may require more edge replication; stronger privacy may reduce utility; robust aggregation may increase protocol cost; and higher accuracy may require larger models or more communication. End-to-end agentic evaluation must therefore connect these system metrics to decision quality, constraint violations, recovery behaviour, and human oversight rather than reporting model accuracy alone.

Active Debates in Federated Agentic Edge Intelligence

Does Federated Learning Provide Sufficient Privacy for Distributed LLM Training?

Federated learning (FL) reduces direct exposure of private corpora because clients transmit updates rather than raw records, but this property does not establish complete privacy. Model parameters, gradients, logits, and generated knowledge can disclose information through inference or reconstruction attacks. The privacy risk is especially relevant for LLMs because updates may encode rare phrases, preferences, or domain-specific records. Surveys of federated LLM research therefore distinguish data locality from formal protection, secure aggregation, differential privacy, and protection against malicious participants [4], [54].

The evidence also shows a utility cost. ROFED-LLM combines split learning, differential privacy, pruning, beamforming, and resource allocation, reporting a 3.94% accuracy trade-off while improving its reported privacy-preservation endpoint by 12.87% [51]. A separate knowledge-distillation study reports that differential privacy can produce up to 70% accuracy loss at privacy budgets of $\varepsilon \leq 10$, although the exact task and configuration determine the magnitude [49]. These results reject the assumption that privacy can be added without changing model quality, communication, or training stability.[55]

Secure aggregation addresses server visibility but does not remove poisoning, collusion, or client-side leakage. P2Brofl combines privacy-preserving three-party computation with Byzantine-robust selection, and its model error increased by 1.05% at 50% Byzantine participants compared with a 23.78% increase without protection [50]. The finding concerns robustness under a stated attack proportion, not a guarantee against all privacy attacks. Future evaluations should report threat model, privacy budget, attack success, utility loss, and protection of both client data and model intellectual property.

Centralized Coordination Versus Decentralized and Multi-Agent Autonomy

The central coordinator remains attractive because it simplifies aggregation, policy enforcement, version control, and global evaluation. Conventional FL commonly uses a server to collect local models, while hierarchical designs add edge aggregators to reduce direct device-server traffic [1], [13]. This structure supports auditable updates and coordinated resource allocation, but it creates dependence on server availability and may become a bottleneck when devices, links, and tasks change rapidly.

Decentralized and multi-agent designs distribute coordination across peers, edge nodes, and cloud services. FedLMA uses decentralized knowledge distillation and an LLM-driven multi-agent mechanism to assess learning difficulty, retention, and client-specific knowledge demand under non-IID data [29]. Hierarchical FL similarly reduces dependence on a single controller through intermediate cluster heads, although it introduces resource-allocation and incentive problems [13]. The question is therefore not whether autonomy should replace coordination, but which decisions require global authority and which can be delegated locally.

Agentic systems add a second layer of uncertainty because an LLM may select tools, formulate plans, or revise actions rather than merely optimize model parameters. KoMA uses interaction, multistep planning, shared memory, and reflection for autonomous-driving decisions, but its evidence concerns framework performance rather than long-term operational safety [7]. A defensible architecture should retain centralized constraints for safety-critical actions while allowing local agents to adapt sensing, caching,

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

scheduling, and model selection. This division preserves accountability without forcing every low-latency decision through a remote coordinator.[56]

Cloud Accuracy Versus Edge Responsiveness and Resource Feasibility

Cloud deployment retains advantages in memory capacity, model scale, and access to centralized computation. Edge inference reduces network dependence and can improve responsiveness, privacy, and bandwidth use, but device memory, energy, and processing limits constrain model size and context length [3], [14]. The relevant comparison is therefore task-specific: a cloud model may produce stronger general reasoning, while a smaller edge model may satisfy a deadline or operate during disconnection.

Collaborative execution offers an intermediate option. EdgeShard partitions Llama2 models across devices and cloud resources, reporting up to 50% lower inference latency and twice the throughput on a real-world testbed without reported accuracy loss relative to its comparison setting [15]. These gains depend on suitable device selection, partition placement, and network conditions. Split training reaches a similar compromise: FedsLLM combines LoRA with split federated learning and reports a 47.63% average reduction in training delay after communication and computation optimization [18].

The literature therefore supports adaptive placement rather than a universal edge-first rule. A system should select local, collaborative, or cloud execution according to deadline, model confidence, energy state, privacy requirement, and link reliability. Claims of superior accuracy or responsiveness must specify whether the endpoint is token quality, task success, latency, throughput, or training convergence. Without this separation, cloud and edge systems appear comparable even when they optimize different operational objectives.

Model Homogeneity Versus Heterogeneous Client Architectures

Homogeneous FL simplifies aggregation because every client trains the same architecture and communicates compatible parameters. That assumption conflicts with real deployments involving unequal memory, processors, battery states, model licenses, and network links. Heterogeneous FL research identifies statistical, model, communication, and device heterogeneity as separate sources of difficulty [5]. LLM deployment intensifies the problem because full-model updates are expensive and client capabilities can differ by orders of magnitude.

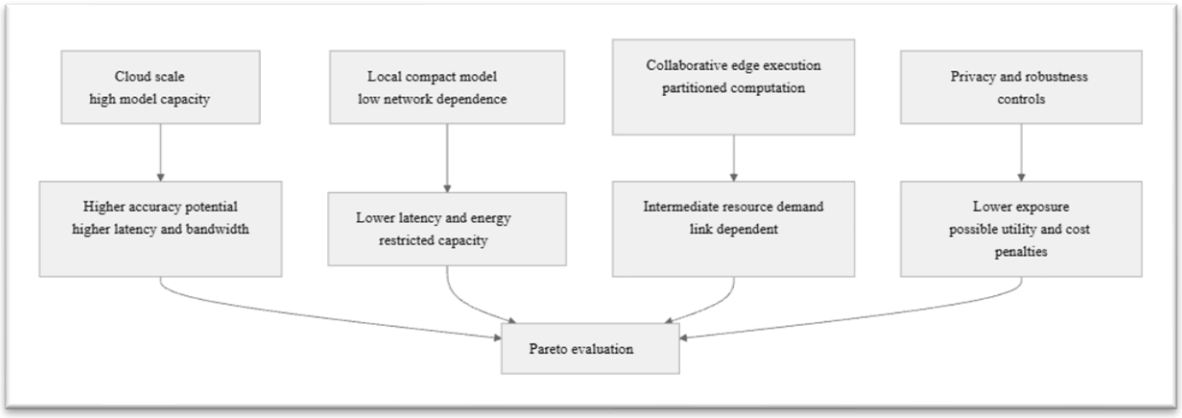
Knowledge transfer permits clients to use different models. FedCache exchanges logits and cached knowledge rather than requiring identical parameter interaction, reporting more than two orders of magnitude higher communication efficiency than its comparison methods while maintaining comparable performance [28]. Fed-ET trains heterogeneous small client models and distils their weighted consensus into a larger server model, reducing communicated parameters and improving performance under high data heterogeneity [26]. These approaches increase architectural flexibility, but knowledge alignment and calibration become additional evaluation requirements.[57]

Compression creates another form of controlled heterogeneity. PruneFL adapts model size during federated training on edge devices while maintaining accuracy similar to the original model [35]. FedDrop assigns subnets with heterogeneous dropout rates according to channel conditions, reducing computation and communication relative to conventional FL [36]. The unresolved issue is whether these methods preserve reasoning, safety behavior, and personalization for LLM agents rather than only classification accuracy.

Graph of the Accuracy–Latency–Communication–Energy Pareto Frontier

The **Pareto frontier** is more appropriate than a single ranking because improvements in one endpoint can worsen another. Pruning, distillation, and local inference may reduce latency and energy while narrowing model capacity; privacy noise may protect updates while reducing accuracy; robust aggregation may increase computation while limiting attack damage. Reported results from ROFED-LLM, EdgeShard, AdaSFL, and FedCache illustrate distinct operating points rather than a common scale [15], [17], [28], [51].

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084



Future studies should plot accuracy or decision quality against latency, communication volume, and energy under identical workloads. The frontier should also include privacy loss, failure recovery, and unsafe-action rates, because an apparently efficient point may be unacceptable for autonomous control. Such evaluation connects system optimization to the evidence gaps addressed next.

Evidence Synthesis and Research Gaps for Autonomous Edge Decision Systems

Convergent Findings: Why LLMs, Federated Learning, and Edge Computing Are Complementary

The literature converges on a functional division. LLMs provide language-mediated intent interpretation, planning, code generation, and knowledge integration [2]. FL permits collaborative adaptation while keeping training data at participating sites, and edge computing places inference and coordination near sensors, users, and actuators [1]. Their combination addresses different constraints: semantic flexibility from LLMs, distributed learning from FL, and responsiveness from edge execution.[58].

This complementarity is conditional rather than automatic. OpenFedLLM reports that federated algorithms outperform local training across its evaluated settings, while FedLLM-Bench argues that earlier evaluations often relied on artificially constructed data and introduces client diversity in language, quality, quantity, instruction, length, embedding, and preference [22], [24]. Thus, federated adaptation can improve shared models, but the result depends on client composition, task alignment, and evaluation realism. Edge placement further requires compression, scheduling, and model partitioning [3].

Underdeveloped Evidence on End-to-End Agentic Decision Quality and Real-World Autonomy

Most reported endpoints remain model accuracy, convergence, latency, throughput, or communication cost. These measures do not establish that an agent perceives correctly, plans safely, uses tools appropriately, recovers from failure, or respects operational constraints. KoMA evaluates multi-agent reasoning for driving, while LEI evaluates resource use during generated device-side logic, but neither abstract establishes long-duration autonomous operation across changing conditions [7], [10]. The gap is larger for actuation, where an incorrect recommendation and an incorrect physical action have different consequences.[59]

Research should therefore evaluate complete perception-reasoning-action loops. Required endpoints include task completion, constraint violations, calibration, intervention frequency, recovery after link loss, and degradation under stale or conflicting knowledge. Agent traces should be retained so that planning errors can be distinguished from sensing, communication, or execution errors. Without this decomposition, a low-latency answer may be mistaken for a reliable autonomous decision.

Benchmarking Gaps Across Heterogeneous Hardware, Wireless Conditions, and Application Domains

Benchmark evidence is fragmented across simulations, testbeds, datasets, and application-specific tasks. FedLLM-Bench improves comparability for federated LLM training, but its scope does not by itself evaluate heterogeneous accelerators, intermittent wireless links, actuator timing, or multi-agent coordination [24]. Satellite FEEL introduces mobility and short ground-link windows, whereas vehicular, industrial, medical, and agricultural systems impose different data distributions and safety constraints [8], [33]. Cross-domain claims therefore require caution.[60]

A reference benchmark should vary model size, client architecture, memory, battery state, bandwidth, packet loss, mobility, participation, and non-IID severity. It should report accuracy and decision quality together with per-client energy, peak memory, transmitted bytes, staleness, fairness, and failure recovery. Hardware-in-the-loop trials are needed because simulated communication costs can conceal thermal throttling, scheduling contention, and device dropout. Standardized traces and reproducible software pipelines, such as FederatedScope-LLM, can support these comparisons [21].[61]

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

Missing Governance, Interpretability, Safety, and Human Oversight Evaluations
LLM agents can generate explanations, but textual explanations do not prove causal faithfulness or safe behavior. Federated systems also distribute responsibility across data owners, aggregators, model developers, and operators. Current work addresses privacy, Byzantine robustness, and trust-aware orchestration, yet these protections do not define who approves model updates, audits agent actions, or accepts residual risk [43], [50]. Governance must therefore be evaluated as an operational process rather than mentioned as a design objective.

Safety studies should test refusal, uncertainty reporting, escalation, authorization boundaries, and human override under adversarial prompts, corrupted updates, sensor conflict, and communication failure. Interpretability should compare explanations with logged observations, intermediate plans, and executed actions. Human oversight should report intervention burden and the conditions under which operators can understand or reverse an agent decision. These measures would connect technical autonomy to accountable deployment.[62]

Proposed Research Agenda and Reference Evaluation Protocol for Future Systems
A staged agenda should first standardize datasets, hardware profiles, network traces, threat models, and energy measurement. The second stage should evaluate federated adaptation with explicit privacy budgets, poisoning rates, client heterogeneity, and update staleness. The third should place the adapted models inside agentic loops and measure task success, safety violations, recovery, and human intervention. The final stage should use prolonged hardware-in-the-loop or field trials with documented rollback and audit procedures.[62]

A reference protocol can be represented as: **scenario definition → local sensing and inference → federated or decentralized adaptation → agent planning → constrained action → failure injection → audit**. Each run should report distributions, not only averages, because tail latency, worst-client energy, and rare unsafe actions determine deployment risk. Comparisons should include cloud, local, hierarchical, and decentralized baselines with matched model quality targets. This protocol would make the Pareto frontier operationally meaningful rather than merely computational.

II. CONCLUSION

Integrated Design Principles for Autonomous Decision Systems
Trustworthy agentic edge intelligence should combine local data control, resource-aware model placement, adaptive federation, and explicit action constraints. LLMs can interpret intent and coordinate heterogeneous models, FL can support distributed personalization, and edge execution can reduce dependence on remote connectivity. None of these properties guarantees safety independently.

Designs should separate reversible recommendations from safety-critical actuation, preserve human escalation, and expose uncertainty and provenance. Privacy mechanisms, robust aggregation, compression, and scheduling must be assessed jointly because each can alter accuracy, latency, energy, or participation. Evaluation should therefore target decision quality and failure behavior alongside conventional model metrics.

Conditions Required for Translation from Prototype Architectures to Deployed Edge Ecosystems
Translation requires reproducible benchmarks, heterogeneous hardware trials, realistic wireless traces, formal threat models, and long-duration operational evidence. Systems must define update authorization, audit logs, rollback procedures, and responsibility for agent decisions. They must also demonstrate acceptable tail latency, energy sustainability, privacy loss, robustness, and human intervention burden under degraded conditions.

The central research task is to replace isolated demonstrations with accountable system evidence. Deployment becomes credible when an agentic edge architecture can adapt without exposing sensitive data, remain useful across unequal clients, explain and constrain its actions, and fail safely when computation, communication, or supervision is unavailable.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

REFERENCES

[1]. W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020, doi: 10.1109/comst.2020.2986024.

[2]. C. K. Dehury, S. S. Kushwaha, Q. Zhang, A. Saleh, and P. K. Donta, "LLM-assisted agentic edge intelligence framework," *arXiv (Cornell University)*. Mar. 11, 2026. doi: 10.48550/arxiv.2604.09607.

[3]. R. Haider, "From search visibility to answer share: Developing and testing a causal theory of brand discoverability and consumer conversion in generative engine optimization," Jul. 2026, doi: 10.13140/rg.2.2.24339.64807.

[4]. Y. Ye, L. Shen, F. Liu, Y. Tang, and W. Hu, "EdgeFed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020, doi: 10.1109/access.2020.3038287.

[5]. W. Y. B. Lim *et al.*, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, Jul. 2021, doi: 10.1109/tpds.2021.3096076.

[6]. E. Kristiani, V. K. Verma, and Y. Chao-Tung, "Deploying LLM transformer on edge computing devices: A survey of strategies, challenges, and future directions," *AI*, vol. 7, no. 1, p. 15, Jan. 2026, doi: 10.3390/ai7010015.

[7]. M. Zhang, X. Shen, J. Cao, Z. Cui, and S. Jiang, "EdgeShard: Efficient LLM inference via collaborative edge computing," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 13119–13131, Dec. 2024, doi: 10.1109/jiot.2024.3524255.

[8]. R. Haider, "Attention sovereignty in digital markets: Designing and testing consumer-owned preference agents as an alternative to surveillance-based advertising," Jul. 2026, doi: 10.13140/rg.2.2.27695.09125.

[9]. Y. Liao, Y. Xu, H. Xu, Z. Yao, L. Wang, and C. Qiao, "Accelerating federated learning with data and model parallelism in edge computing," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 904–918, Aug. 2023, doi: 10.1109/tnet.2023.3299851.

[10]. K. Zhao, Z. Yang, C. Huang, X. Chen, and Z. Zhang, "FedsLLM: Federated split learning for large language models over communication networks," in *2024 International Conference on Ubiquitous Communication (Ucom)*, IEEE, Jul. 2024, pp. 438–443. doi: 10.1109/ucom62433.2024.10695888.

[11]. K. Zhao, Z. Yang, Y. Hu, M. Chen, C. Zhu, and Z. Zhang, "Efficient split federated learning for large language models over communication networks," *arXiv (Cornell University)*. Apr. 20, 2025. doi: 10.48550/arxiv.2504.14667.

[12]. Y. Shen *et al.*, "Large language models empowered autonomous edge AI for connected intelligence," *IEEE Communications Magazine*, vol. 62, no. 10, pp. 140–146, Jan. 2024, doi: 10.1109/mcom.001.2300550.

[13]. J. Rani, M. Rana, S. Kumar, and P. Prasad, "EASA: Distributed agentic framework for scalable federated edge intelligence," in *2026 International Conference on Intelligent Computing and Automation for Sustainable Solutions (ICASS)*, IEEE, Feb. 2026, pp. 1–7. doi: 10.1109/icass69550.2026.11547499.

[14]. W. Kuang *et al.*, "FederatedScope-LLM: A comprehensive package for fine-tuning large language models in federated learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2024, pp. 5260–5271. doi: 10.1145/3637528.3671573.

[15]. R. Ye *et al.*, "OpenFedLLM: Training large language models on decentralized private data via federated learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2024, pp. 6137–6147. doi: 10.1145/3637528.3671582.

[16]. K. Zhao, C. Zhu, M. Chen, C. Huang, Z. Yang, and Z. Zhang, "SflLLM: Efficient split federated learning for large language model over wireless networks," in *GLOBECOM 2025 - 2025 IEEE Global Communications Conference*, IEEE, Dec. 2025, pp. 1835–1840. doi: 10.1109/globecom59602.2025.11432069.

[17]. R. Ye *et al.*, "FedLLM-Bench: Realistic benchmarks for federated learning of large language models," *arXiv (Cornell University)*. Jun. 07, 2024. doi: 10.48550/arxiv.2406.04845.

[18]. "European journal of prosthodontics and restorative dentistry," May 2026, doi: 10.1922/ejprd.v34i.

[19]. Y. J. Cho, A. Manoel, G. Joshi, R. B. Sim, and D. Dimitriadis, "Heterogeneous ensemble knowledge transfer for training large models in federated learning," *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 2881–2887, Jul. 2022, doi: 10.24963/ijcai.2022/399.

[20]. L. Fang *et al.*, "Knowledge distillation and dataset distillation of large language models: emerging trends, challenges, and future directions," *Artificial Intelligence Review*, vol. 59, no. 1, p. 17, Nov. 2025, doi: 10.1007/s10462-025-11423-3.

[21]. Z. Wu *et al.*, "FedCache: A knowledge cache-driven federated learning architecture for personalized edge intelligence," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9368–9382, Feb. 2024, doi: 10.1109/tmc.2024.3361876.

[22]. J. Wang *et al.*, "FedLMA: A federated learning framework integrating LLM-Based multi-agent reasoning with knowledge distillation," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 4, pp. 11339–11349, Sep. 2025, doi: 10.1109/tce.2025.3608003.

[23]. G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 6, pp. 3820–3860, Jan. 2025, doi: 10.1109/comst.2025.3527641.

[24]. R. Haider *et al.*, "Forensic consumer psychology: An evidence-based framework for detecting deception and digital market manipulation," 2026.

[25]. K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, Jan. 2020, doi: 10.1109/twc.2019.2961673.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

[26]. X. Cao, G. Zhu, J. Xu, Z. Wang, and S. Cui, "Optimized power control design for over-the-air federated edge learning," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 342–358, Nov. 2021, doi: 10.1109/jsac.2021.3126060.
[27]. M. Zhang *et al.*, "Communication-Efficient federated edge learning via optimal probabilistic device scheduling," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8536–8551, Apr. 2022, doi: 10.1109/twc.2022.3166941.
[28]. T. Wang, Y. Liu, X. Zheng, H. Dai, W. Jia, and M. Xie, "Edge-Based communication optimization for distributed federated learning," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2015–2024, Jun. 2021, doi: 10.1109/tnse.2021.3083263.
[29]. Y. Jiang *et al.*, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10374–10386, Apr. 2022, doi: 10.1109/tnnls.2022.3166101.
[30]. D. Wen, K. J. Jeon, and K. Huang, "Federated dropout—A simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 923–927, Feb. 2022, doi: 10.1109/lwc.2022.3149783.
[31]. R. Haider, Md. R. Amin, Md. S. Arafat, I. Hossain, and R. Ahmad, "Smart Automation: Revolutionizing Business Operations, Advertising Costs and Customer Satisfaction Through Technology," 2026.
[32]. M. Ma, C. Gong, L. Zeng, Y. Yang, and L. Wu, "FlocOff: Data heterogeneity resilient federated learning with communication-efficient edge offloading," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 11, pp. 3262–3277, Jul. 2024, doi: 10.1109/jsac.2024.3431526.
[33]. Q. Chen, Z. You, D. Wen, and Z. Zhang, "Enhanced hybrid hierarchical federated edge learning over heterogeneous networks," *IEEE Transactions on Vehicular Technology*, pp. 1–15, 2023, doi: 10.1109/tvt.2023.3287355.
[34]. F. Piccialli, D. Chiaro, P. Qi, V. Bellandi, and E. Damiani, "Federated and edge learning for large language models," *Information Fusion*, vol. 117, p. 102840, Dec. 2024, doi: 10.1016/j.inffus.2024.102840.
[35]. H. Yang, Z. Wang, and L. Cui, "Energy-Efficient federated learning in mobile edge computing via fine-grained energy planning client selection," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2026, doi: 10.1109/tmc.2026.3668371.
[36]. RaiyanHaider, "Generative AI and quantum-inspired optimization: Redefining portfolio risk management and real-time capital allocation in volatile markets," *Journal of Informatics Education and Research*, Mar. 2026, doi: 10.52783/jier.v6i1.4540.
[37]. R. Haider, T. Dwivedi, A. G. G. Girish, N. Verma, B. Kashyap, and V. S. Dubey, "Neuromarketing approaches to shaping healthy consumer choices: An integrative analysis of methods, efficacy, and ethical considerations," *CLOK (University of Central Lancashire)*, Mar. 2026, doi: 10.25258/ijddt.16.4s.62.
[38]. B. Das, N. Ali, S. Pirbhulal, G. Aloi, P. Pace, and A. H. Sodhro, "Adaptive federated learning for 6g: A multi-agent architecture for 6g edge intelligence," *IEEE Network*, pp. 1–8, 2026, doi: 10.1109/mnet.2026.3694694.
[39]. B. Jiang, J. Li, H. Wang, and H. Song, "Privacy-Preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1136–1144, Nov. 2021, doi: 10.1109/tii.2021.3131175.
[40]. X. Zheng *et al.*, "Exploring multi-agent dynamics for generative AI and large language models in mobile edge networks," *IEEE Wireless Communications*, vol. 32, no. 6, pp. 69–78, Oct. 2025, doi: 10.1109/mwc.2025.3599602.
[41]. R. Haider, Md. F. I. Shaif, R. Ahmed, N. H. Nafi, M. R. Sumon, and M. Rahman, "Assessing the impact of influencer marketing on brand value and business revenue: An empirical and thematic analysis," *International Journal of Science and Research Archive*, vol. 16, no. 2, pp. 471–482, Aug. 2025, doi: 10.30574/ijsra.2025.16.2.2355.
[42]. B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-Effective federated learning in mobile edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3606–3621, Oct. 2021, doi: 10.1109/jsac.2021.3118436.
[43]. R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 148–155, 2021, doi: 10.1109/mnet.011.2000295.
[44]. G. Gad, E. Gad, Z. Md. Fadlullah, M. M. Fouda, and N. Kato, "Communication-Efficient and privacy-preserving federated learning via joint knowledge distillation and differential privacy in bandwidth-constrained networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17586–17601, Jul. 2024, doi: 10.1109/tvt.2024.3423718.
[45]. M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, Sep. 2023, doi: 10.1145/3625558.
[46]. C. Dong *et al.*, "Privacy-Preserving and byzantine-robust federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 889–904, Apr. 2023, doi: 10.1109/tdsc.2023.3264697.
[47]. H. Wang *et al.*, "ROFED-LLM: Robust federated learning for large language models in adversarial wireless environments," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 1084–1096, Jul. 2025, doi: 10.1109/tnse.2025.3590975.
[48]. R. Haider, Md. F. I. Shaif, R. Ahmed, N. H. Nafi, M. R. Sumon, and M. Rahman, "The influence of social media branding on consumer purchase behavior: A comprehensive empirical and thematic analysis," *International Journal of Science and Research Archive*, vol. 16, no. 2, pp. 460–470, Aug. 2025, doi: 10.30574/ijsra.2025.16.2.2354.
[49]. A. Tak and S. Cherkaoui, "Federated edge learning: Design issues and challenges," *IEEE Network*, vol. 35, no. 2, pp. 252–258, Oct. 2020, doi: 10.1109/mnet.011.2000478.
[50]. C. Chen *et al.*, "Integration of large language models and federated learning," *Patterns*, vol. 5, no. 12, p. 101098, Dec. 2024, doi: 10.1016/j.patter.2024.101098.

ISSN :2584-2706 DOI: https://doi.org/10.67317/IJMSRT26OCT084

[51]. R. Haider, J. T. Juba, and S. Chowdhury, "Exploring the link between suicidal ideation and digital environments: The hidden impact of marketing content," *International Journal of Science and Research Archive*, vol. 16, no. 2, pp. 607–614, Aug. 2025, doi: 10.30574/ijsra.2025.16.2.2353.

[52]. R. Haider, W. A. Megha, J. T. Juba, A. Alamgir, and L. Ahmad, "The conversational revolution in health promotion: Investigating chatbot impact on healthcare marketing, patient engagement, and service reach," *International Journal of Science and Research Archive*, vol. 15, no. 3, pp. 1585–1592, Jun. 2025, doi: 10.30574/ijsra.2025.15.3.1937.

[53]. R. Haider, F. A. I. Bari, Osru, N. Afia, and M. A. khan Mugdho, "Leveraging internet of things data for real-time marketing: Opportunities, challenges, and strategic implications," *International Journal of Science and Research Archive*, vol. 15, no. 3, pp. 1657–1663, Jun. 2025, doi: 10.30574/ijsra.2025.15.3.1936.

[54]. R. Haider, M. Bari, Md. F. I. Shaif, M. Rahman, Md. N. H. Ohi, and K. M. M. Rahman, "Quantifying the impact: Leveraging AI-Powered sentiment analysis for strategic digital marketing and enhanced brand reputation management," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 1103–1121, May 2025, doi: 10.30574/ijsra.2025.15.2.1524.

[55]. R. Haider, "Navigating the digital political landscape: How social media marketing shapes voter perceptions and political brand equity in the 21st century," *International Journal of Science and Research Archive*, vol. 15, no. 1, pp. 1736–1744, Apr. 2025, doi: 10.30574/ijsra.2025.15.1.1217.

[56]. J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–41, Apr. 2024, doi: 10.1145/3659205.

[57]. R. Haider, M. Bari, Osru, N. Afia, and T. Al Karim, "Illuminating the black box: Explainable AI for enhanced customer behavior prediction and trust," *International Journal of Science and Research Archive*, vol. 15, no. 3, pp. 247–268, Jun. 2025, doi: 10.30574/ijsra.2025.15.3.1674.

[58]. R. Haider and J. Sabatina, "Harnessing the power of micro-influencers: A comprehensive analysis of their effectiveness in promoting climate adaptation solutions," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 595–610, May 2025, doi: 10.30574/ijsra.2025.15.2.1448.

[59]. R. Haider, M. Bari, Md. F. I. Shaif, and M. Rahman, "Engineering hyper-personalization: Software challenges and brand performance in AI-driven digital marketing management: An empirical study," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 1122–1141, May 2025, doi: 10.30574/ijsra.2025.15.2.1525.

[60]. K. Jiang *et al.*, "KoMA: Knowledge-Driven multi-agent framework for autonomous driving with large language models," *IEEE Transactions on Intelligent Vehicles*, vol. 10, no. 10, pp. 4655–4668, Nov. 2024, doi: 10.1109/tiv.2024.3488793.

[61]. Y. Shi, L. Zeng, J. Zhu, Y. Zhou, C. Jiang, and K. B. Letaief, "Satellite federated edge learning: Architecture design and convergence analysis," *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 15212–15229, Jul. 2024, doi: 10.1109/twc.2024.3427377.

[62]. R. Haider, "The algorithmic memory of brands: Developing a longitudinal theory of how large language models learn, retain, distort and correct brand knowledge proposal field details," Jul. 2026, doi: 10.13140/rg.2.2.26017.36966.