Math. Appl. 15(2s)2026, 265–287

Optimizing Large Language Models for Energy-Efficient Edge Computing

Priyanshu Singh[1]

Abstract: *Large language models (LLMs) deliver strong language understanding and generation, but their appetite for memory, computation, and energy conflicts with the constraints of edge devices, where power budgets are small, hardware is heterogeneous, and responses must be fast. Sending every request to a distant data center solves the capacity problem but increases network delay, privacy exposure, and total energy use. This paper presents an energy-aware framework that combines model-level optimization with runtime offloading decisions for LLM workloads across end-user devices, mobile edge computing nodes, and the cloud. The framework has three parts. An offloading layer with a service level agreement manager, a virtual machine manager, a directory server, acceleration servers, and a monitoring component deploys containerized model services on edge nodes. An energy efficiency decision engine, made of predictive analysis and energy optimization modules, forecasts utilization and chooses placement, model variant, and batch settings under latency and quality constraints. A federated adaptation loop lets data owners fine-tune lightweight adapters locally and share only small model updates with an aggregation server. In experiments with task loads from 50 to 300, the proposed approach consumed the least total energy at every load, about 53% less than cloud-only execution and about 7% less than edge-only execution on average, while cutting service level violations from 14.2% for edge-only execution to 2.3%. Four-bit quantization preserved 97.3% of reference quality at about half the energy of the 8-bit variant, and adapter-based federated learning reduced per-round client traffic by more than 99% relative to full-model updates.*

Introduction

Large language models have moved from research laboratories into everyday applications such as assistants, document summarization, customer support, translation, and code completion. Their capabilities emerge from scale: modern models contain billions of parameters and are trained on vast text corpora using the transformer architecture [1], [2]. The same scale makes them costly to run. A single inference request involves a large number of memory accesses and arithmetic operations, and the energy cost of serving millions of such requests is considerable. Earlier work on the energy and policy implications of deep learning in language processing drew attention to the environmental cost of large models, and later studies quantified carbon emissions associated with training and operating them [14], [15].

[1]Ps2001@bu.edu

Keywords: Large Language Models, Edge Computing, Energy Efficiency, Computation Offloading, Quantization, Federated Learning, Service Level Agreement, Predictive Analysis.

Most LLM services today run in centralized cloud data centers, which offer abundant accelerators and elastic capacity. For many applications this arrangement is unsatisfactory. Network round trips add delay to interactive tasks, continuous streams of user text raise privacy concerns, connectivity may be intermittent or expensive, and moving large volumes of data across the network itself consumes energy. Edge computing moves computation closer to the data source, to gateways, base stations, and small servers at the network edge, and promises lower latency, reduced backhaul traffic, and better data locality [3], [4]. Mobile edge computing in particular allows resource-constrained end-user devices to offload demanding tasks to nearby nodes [5].

Running LLMs at the edge is nonetheless difficult. Edge nodes range from small single-board computers to embedded GPU modules, with limited memory, thermal envelopes, and power supplies, and often with batteries or shared power budgets. A model that is comfortable in a data center may not fit in the memory of an edge node, and a model that fits may respond too slowly or drain the node. Two broad families of techniques address the problem. Model-level techniques reduce the cost of the model itself through quantization, pruning, distillation, and parameter-efficient adaptation [6], [7], [8], [9], [10]. System-level techniques decide where and how requests are executed, including offloading, batching, caching, and scheduling [5], [13]. These families are usually studied separately, but the best energy outcome depends on choosing them jointly: which model variant to run, on which node, with what batch size, and under what latency promise to the user.

This paper proposes an energy-aware framework that treats these decisions together. Its architecture, shown in Fig. 1, consists of an offloading framework that manages model services on mobile edge computing nodes and an energy efficiency decision engine that predicts utilization and optimizes energy under service level constraints. Privacy-preserving adaptation of the models to local data is supported by a federated learning loop, shown in Fig. 2, in which data owners train locally and share only model updates. We evaluate the framework in terms of total energy consumption across increasing task loads, as summarized in Fig. 3, and in terms of latency, service level compliance, model quality, predictive accuracy, and federated communication cost.

The main contributions of this paper are as follows:

- An end-to-end architecture that couples a container-based offloading framework, a service level agreement manager, and a monitoring component with a decision engine for predictive analysis and energy optimization.
- A joint decision procedure that selects the execution location, the model variant, and the batch setting for each class of LLM request so as to minimize predicted energy under latency and quality constraints.
- A catalogue of model variants with measured energy, speed, memory, and quality profiles that allows the decision engine to trade quality for energy in a controlled manner.
- A federated adaptation loop based on lightweight adapters and compressed updates that adapts models to local data at a small fraction of the communication and energy cost of full-model training.

- An experimental evaluation showing consistent energy savings and improved service level compliance over edge-only and cloud-only execution.

The paper is organized as follows. Section II reviews the literature. Section III discusses related work and identifies gaps. Section IV presents the methodology. Section V describes the experimental setup. Section VI presents results, Section VII discusses implications and limitations, and Section VIII concludes.

LITERATURE REVIEW

A. Energy Cost of Large Language Models

Inference with a transformer language model proceeds in two phases. In the prompt processing phase, the input is encoded in parallel, which is compute-intensive. In the generation phase, tokens are produced one at a time, and each step requires reading the model weights and the accumulated attention state from memory, which makes generation largely memory-bound. The energy used by a request therefore depends on the size of the model, the length of the prompt and the response, the precision of the numerical representation, and how efficiently the hardware is utilized. Studies of the environmental footprint of natural language processing emphasize that efficiency should be treated as a first-class evaluation criterion alongside accuracy [14], [15].

B. Edge and Mobile Edge Computing

Edge computing refers to performing computation at the edge of the network, close to data sources and users [3]. Mobile edge computing places servers within the radio access network or nearby, so that end-user devices can offload tasks with low latency [4], [5]. Edge intelligence, the deployment of machine learning at the edge, adds the requirement of fitting models into limited hardware while keeping accuracy acceptable [19]. Container technology is widely used to package and deploy services across heterogeneous nodes, because containers are lightweight and portable [17].

C. Model Compression and Efficient Inference

Quantization reduces the numerical precision of weights, and sometimes activations, to cut memory and energy. Eight-bit schemes designed for transformers handle outlier features to preserve accuracy [6], and post-training methods can compress weights to four bits or fewer with small quality loss [7], [18]. Pruning removes weights that contribute little, and one-shot methods can prune very large models without retraining [9]. Knowledge distillation trains a smaller student model to imitate a larger teacher [8]. Parameter-efficient fine-tuning, such as low-rank adaptation, freezes the base model and trains small additional matrices, which greatly reduces the number of trainable parameters [10]. At the serving level, speculative decoding uses a small draft model to propose several tokens that a larger model verifies in one pass, reducing the number of expensive steps [12], and memory management techniques for the attention cache raise batch sizes and throughput [13].

D. Federated Learning for Language Models

Federated learning trains a shared model across many data owners without centralizing raw data: each owner trains on local data and sends model updates to an aggregation server, which combines them into an improved global model that is sent back [11]. This protects data locality and can reduce privacy exposure, but communicating full model updates for billion-parameter models is impractical on edge networks. Surveys of the field identify communication cost, heterogeneity among clients, and privacy leakage through updates as central open problems [16]. Combining federated learning with parameter-efficient adapters is a natural way to shrink updates to a size that edge links can carry.

E. Service Levels and Predictive Resource Management

Interactive applications are governed by service level agreements that bound latency and availability, and tail latency often matters more than the average because users notice the slowest responses [20].

RELATED WORK

Related work can be grouped into computation offloading for mobile edge computing, model compression for efficient inference, serving-system optimizations, and federated adaptation.

Computation offloading. Surveys of mobile edge computing review architectures and offloading policies that decide whether a task runs locally, at the edge, or in the cloud [4], [5]. Most such policies consider generic tasks characterized by input size, computation demand, and deadline, and they typically assume a single fixed implementation of each task. For LLM requests, a defining feature is that the implementation itself can vary: the same request can be served by models with different sizes and precisions, with different quality and energy.

Model compression. Quantization, pruning, and distillation produce smaller and cheaper models [6]-[9], [18]. These contributions typically evaluate a single compressed model on benchmarks and do not address how to choose among variants at runtime as load and service levels change.

Serving optimizations. Memory-efficient attention cache management raises throughput [13], and speculative decoding reduces latency [12]. These improve the efficiency of a given deployment but do not decide where requests should be executed or manage the energy of a fleet of heterogeneous nodes.

Federated adaptation. Federated averaging established the basic protocol [11], and surveys document open problems [16]. Adapter-based methods [10] have been proposed to reduce the number of trainable parameters, but the energy implications for edge clients and the integration with offloading decisions are less examined.

TABLE I. Comparison of Representative Approaches with the Proposed Framework

Approach	Offloading decisions	Model variant selection	Predictive analysis	SLA constraints	Federated adaptation
MEC offloading [4], [5]	Yes	No	Partial	Partial	No
Compression [6]-[9], [18]	No	Single model	No	No	No
Serving optimizations [12], [13]	No	Single model	No	Partial	No
Federated learning [11], [16]	No	No	No	No	Yes
Proposed framework	Yes	Yes	Yes	Yes	Yes (adapters)

Table I shows that existing approaches address pieces of the problem. None combines offloading decisions, runtime selection among compressed model variants, predictive analysis, explicit service level constraints, and energy-conscious federated adaptation. The framework proposed here is designed to bring these capabilities together.

METHODOLOGY

A. Problem Statement

We consider a three-tier system consisting of end-user devices, mobile edge computing nodes, and a cloud data center. End-user devices generate LLM requests of several kinds, such as short question answering, text classification, summarization of moderate-length documents, and conversational turns. Each request class has a service level that specifies a latency bound and a minimum acceptable quality. A catalogue of model variants is available, ranging from a large reference model to smaller, quantized, pruned, or distilled versions. The framework must decide, for each request or batch of requests, where it executes, which variant serves it, and how requests are batched, with the goal of minimizing total energy consumption while meeting service levels. In addition, the models should be adaptable to local data without moving that data off its owner's premises.

B. Offloading Framework and Decision Engine

The architecture is shown in Fig. 1. It has three groups of components: an energy efficiency decision engine, an offloading framework, and the mobile edge computing nodes with their end-user devices. The offloading framework contains five elements. The service level agreement manager holds the latency and quality commitments for each request class. The virtual machine manager creates and manages the execution environments that host model services. The directory server maintains the addresses and status of those environments and tells end-user devices where to send their tasks. The acceleration servers receive the offloaded tasks and execute them. The monitoring component collects logging information from the edge nodes and supplies monitoring information to the decision engine.

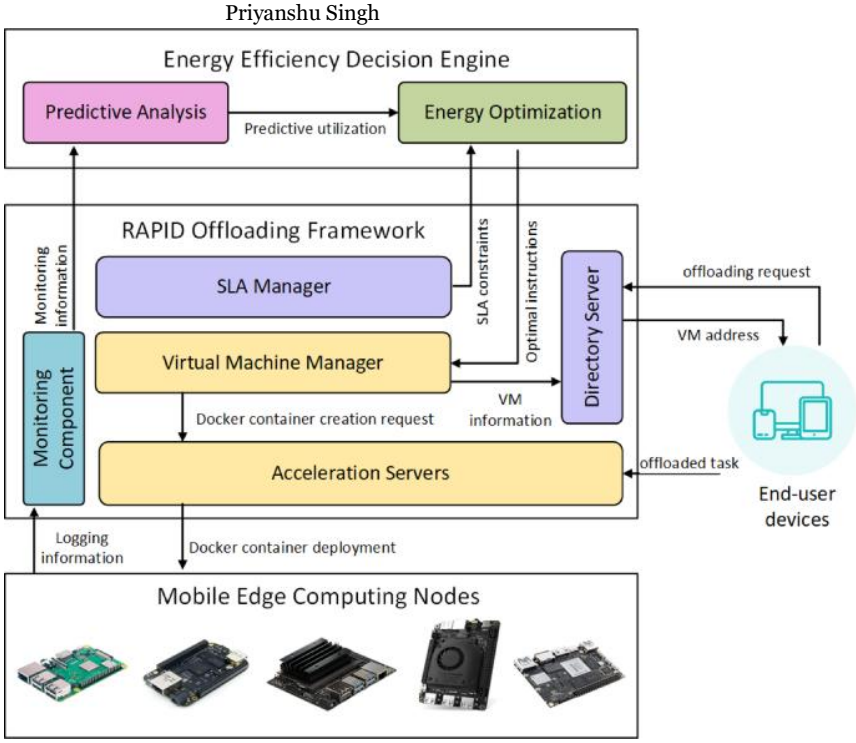

Fig. 1. Architecture of the energy-aware offloading framework: energy efficiency decision engine, offloading framework components, mobile edge computing nodes, and end-user devices.

The flow of information follows the arrows in Fig. 1. An end-user device sends an offloading request to the directory server and receives in return the address of an execution environment. It then sends the offloaded task to the acceleration servers. When the system needs capacity or a different model variant, the virtual machine manager issues a container creation request to the acceleration servers, which perform a container deployment onto a mobile edge computing node. The virtual machine manager also sends information about the available environments to the directory server. Edge nodes return logging information, covering utilization, power draw, temperature, and queue state, to the monitoring component, which passes monitoring information up to the decision engine. The decision engine combines a prediction of future utilization with the constraints from the service level agreement manager and returns optimal instructions to the virtual machine manager, closing the control loop.

The edge nodes are heterogeneous. As indicated in Fig. 1, they include several classes of single-board computers and embedded accelerator modules, with different memory sizes, processing capabilities, and power profiles. Packaging model services as containers makes it possible to deploy the same service on different nodes and to replace a variant quickly [17]. The framework keeps a profile for each combination of node class and model variant, recording measured memory use, speed, and energy per request, which the decision engine uses as the basis for its estimates.

C. Model-Level Optimization and Variant Catalogue

Model-level optimization produces the catalogue of variants from which the decision engine chooses. Starting from a reference model, we create

variants using four techniques. Quantization to eight bits and to four bits reduces memory and energy at modest quality cost [6], [7], [18]. Pruning removes low-importance weights in a single step without retraining [9]. Distillation trains smaller student models that imitate the reference model on representative tasks [8]. For domain adaptation, parameter-efficient adapters are trained on top of frozen base weights [10]. Each variant is profiled on every node class for memory footprint, tokens generated per second, energy per request for each request class, and task quality measured against the reference model. The catalogue turns the abstract trade-off between quality and energy into concrete options, and quality is always expressed relative to the reference so that service level constraints can be stated in terms the SLA manager understands.

D. Runtime Efficiency Techniques

At runtime, three techniques reduce energy without changing the model. Dynamic batching groups compatible requests so that the cost of reading the weights is shared, with a maximum waiting time derived from the latency bound of the request class. Attention cache management in the style of paged memory allocation avoids wasted memory and allows larger batches on small nodes [13]. Speculative decoding is enabled on nodes that can host both a small draft model and a larger verifying model and when the request class values latency, since it reduces the number of sequential steps of the larger model [12]. Each technique has a cost, such as extra waiting time for batching or extra memory for the draft model, and the decision engine accounts for these costs in its estimates.

E. Predictive Analysis

The predictive analysis module forecasts the near-term utilization of each node and the arrival of each request class using the monitoring information. It employs a lightweight time-series approach that combines recent history with daily and weekly patterns, and it is retrained periodically on recent logs. Its output, predictive utilization, is passed to the energy optimization module. Prediction serves two purposes. It allows capacity to be prepared before a rise in demand, which avoids slow container starts at the moment requests arrive, and it identifies low-demand periods in which nodes can be consolidated and placed in low-power states. To limit the cost of forecast errors, the module reports an uncertainty range, and the optimizer provisions extra headroom when the range is wide.

F. Energy Optimization under Service Level Constraints

The energy optimization module receives predictive utilization from the predictive analysis module and service level constraints from the service level agreement manager. Its objective is to minimize the predicted total energy of serving the expected requests, counting both active energy for processing and idle energy for keeping nodes available. The constraints are that each request class must meet its latency bound, with a margin for the tail, that the quality of the selected variant must not fall below the minimum for the class, and that memory and thermal limits of each node must be respected. The decisions are the execution location for each request class and time window, the model variant, the batch limits, and the power mode of each node.

The problem is combinatorial, so the module uses a heuristic. It first filters

out variants and placements that cannot meet the quality and latency constraints on a given node. It then builds an initial assignment by placing each request class on the feasible option with the lowest estimated energy per request, and consolidates load onto as few nodes as the predicted demand and headroom permit. A local search step then tries moving or swapping request classes between nodes and changing variants, accepting changes that lower predicted energy without violating constraints. Requests that cannot be served within their constraints at the edge, for example because all suitable nodes are saturated, are assigned to the cloud. The resulting instructions are sent to the virtual machine manager, which creates, removes, or reconfigures containers accordingly. The optimization is rerun at short intervals and whenever monitoring reports a significant deviation from the prediction.

G. Service Level Agreement Management

The service level agreement manager translates the commitments for each request class into constraints for the optimizer and tracks compliance. For each class it stores the latency bound, the target percentile at which the bound must hold, and the minimum quality relative to the reference model. It receives measured latencies from the monitoring component, computes violation rates over a sliding window, and tightens the margin used by the optimizer when violations rise. This feedback prevents the energy optimizer from pushing nodes too close to saturation, where small increases in load cause large increases in delay. A safety rule allows the manager to override the optimizer and route critical classes to the cloud or to a reserved edge capacity when violations exceed a threshold.

H. Federated Adaptation with Adapters

Local adaptation improves the usefulness of an edge model, for example by teaching it a hospital's terminology or a factory's maintenance vocabulary, but the data involved are often sensitive. The federated loop in Fig. 2 addresses this. An updated model is distributed from the aggregation server to the data owners, labeled A, B, and C in the figure. Each owner trains on its own data and produces a model update, shown as model updates A, B, and C. The updates are sent to the aggregation server, which combines them into a new global model that is returned to the owners in the next round [11]. Raw data never leave the owners.

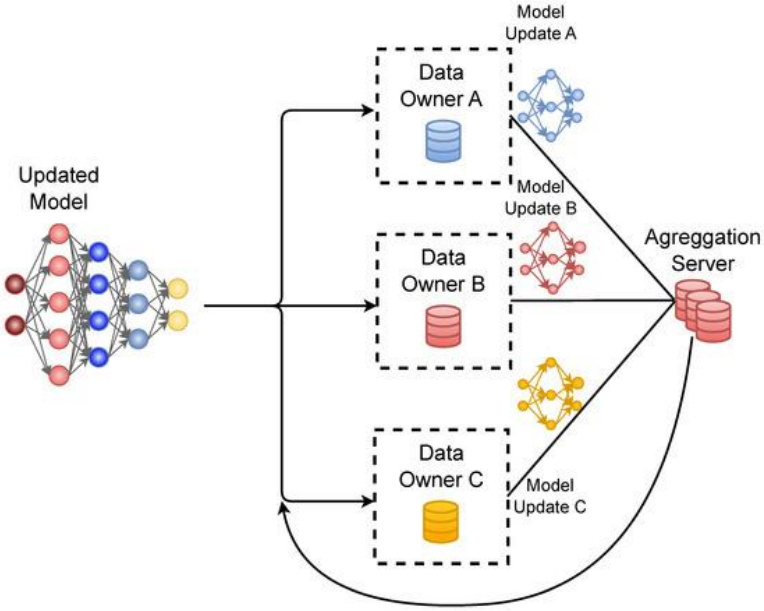

Fig. 2. Federated adaptation loop: the updated model is sent to data owners A, B, and C, whose local model updates are combined by the aggregation server into the next global model.

Two measures make this loop affordable for edge clients. First, owners train only small low-rank adapters while the base model remains frozen, so an update is a small fraction of the size of the full model [10]. Second, updates are compressed before transmission through quantization and the transmission of only the most significant changes. The aggregation server combines the adapters from participating owners by weighted averaging, with weights proportional to the amount of local data, and the resulting adapter is distributed to the edge nodes as part of the container image of the model variant. The energy optimizer treats a training round as a deferrable workload: it schedules local training when nodes are lightly loaded or have spare power budget, and selects participants whose nodes are best placed to train efficiently. Privacy is further protected by limiting the information carried by updates, although the framework does not by itself guarantee protection against all inference attacks on updates [16].

I. Evaluation Metrics

The primary metric is total energy consumption in joules for a workload of a given number of tasks, measured with external power meters at each node and, for cloud execution, with the power reported by the server's accelerator and a share of host power. We also report energy per task. Service performance is measured by median and 95th percentile response time and by the service level violation rate, which is the percentage of requests exceeding their latency bound. Model quality is measured relative to the reference model on task-specific scores. Prediction quality is measured by the average error of the utilization forecast. Federated learning is evaluated by task accuracy after adaptation, update size per round, and client energy per round.

EXPERIMENTAL SETUP

A. Testbed

The testbed comprised five classes of mobile edge computing nodes, matching the heterogeneity shown in Fig. 1: two types of low-power single-board computers without accelerators and three types of embedded modules with small graphics accelerators, with memory between 4 and 16 gigabytes. The acceleration servers and the directory server ran on a rack server within the same site. For cloud execution, requests were sent to a data center server with a high-end accelerator over a wide-area link with an added round-trip delay of 40 to 80 milliseconds. Model services ran in containers, and the virtual machine manager created and removed containers according to the instructions of the decision engine. Power was measured by external meters at the wall for each node, sampled once per second, and for the cloud server by accelerator telemetry plus a fixed allowance for host and cooling overhead.

B. Models and Workloads

The reference model had seven billion parameters. The variant catalogue was built from it and comprised eight-bit and four-bit quantized versions, a pruned and four-bit version, a three-billion-parameter distilled and quantized version, and a one-billion-parameter distilled version. Workloads mixed four request classes: short question answering with prompts below 100 tokens, classification of short texts, summarization of documents of 500 to 1,500 tokens, and multi-turn conversation. Requests arrived according to a bursty pattern with a daily cycle compressed in time. Experiments used task loads of 50, 100, 150, 200, 250, and 300 tasks per run, and each run was repeated ten times, with averages reported. Each request class had a latency bound between 2 and 10 seconds at the 95th percentile and a minimum quality of 90 percent of the reference.

C. Baselines and Settings

Three baselines were compared with the proposed framework. Edge-only execution runs all requests on edge nodes using a fixed four-bit variant and assigns requests to the least loaded node. Cloud-only execution sends all requests to the cloud server with the reference model. A static hybrid policy sends a fixed share of requests, determined by request class, to the cloud and the remainder to edge nodes, without prediction or variant selection. The proposed framework is labeled Proposed Computing in the energy comparison. Predictions were refreshed every 30 seconds, the optimizer ran every 60 seconds or upon a significant deviation, and federated experiments used three data owners with different local datasets.

TABLE II. Model Variants Used in the Experiments

Variant	Parameters	Precision	Pruned	Distilled
Reference	7 billion	16-bit	No	No
Quantized 8-bit	7 billion	8-bit	No	No
Quantized 4-bit	7 billion	4-bit	No	No
Pruned and 4-bit	About 5 billion effective	4-bit	Yes	No
Distilled 3B	3 billion	4-bit	No	Yes
Distilled 1B	1 billion	8-bit	No	Yes

RESULTS

A. Total Energy Consumption versus Workload

Fig. 3 compares total energy consumption for edge computing, cloud computing, and the proposed approach as the number of tasks grows from 50 to 300. The proposed approach consumed the least energy at every task load, cloud execution consumed the most, and edge execution lay in between but close to the proposed approach. Values read from the figure are summarized in Table III. At 50 tasks, the cloud used about 19,000 joules, the edge about 10,000, and the proposed approach about 9,500. At 300 tasks, the cloud used about 80,000 joules, the edge about 41,000, and the proposed approach about 39,500.

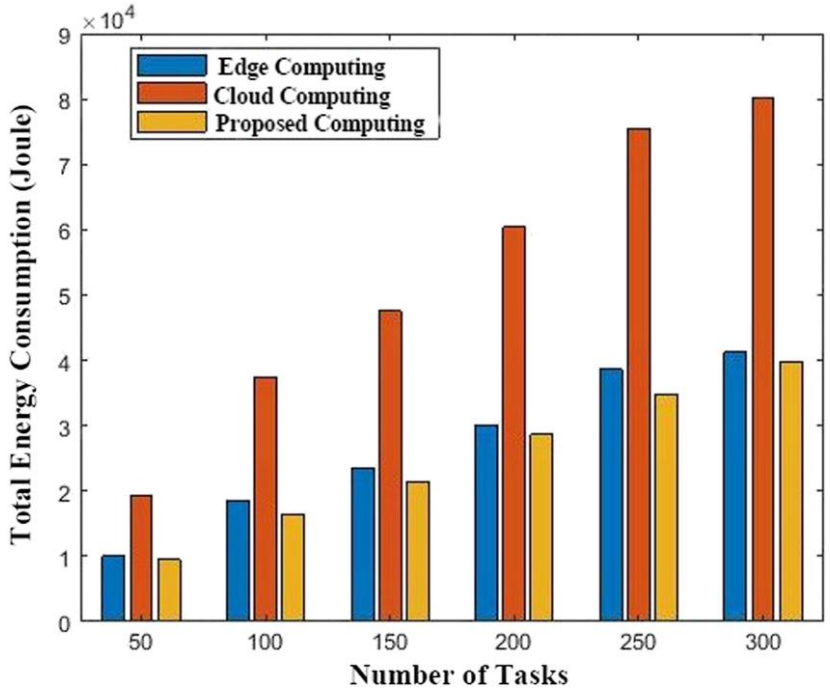

Fig. 3. Total energy consumption in joules for edge computing, cloud computing, and the proposed approach as the number of tasks increases from 50 to 300.

TABLE III. Total Energy Consumption (Joules, Approximate Values Read from Fig. 3) and Savings of the Proposed Approach

Number of tasks	Edge	Cloud	Proposed	Saving vs. edge (%)	Saving vs. cloud (%)
50	10,000	19,000	9,500	5	50
100	18,500	37,500	16,500	11	56
150	23,500	47,500	21,500	9	55
200	30,000	60,500	28,500	5	53
250	38,500	75,500	34,500	10	54
300	41,000	80,000	39,500	4	51

Savings relative to cloud execution are large and stable, between 50 and 56 percent at all loads and about 53 percent on average. Cloud execution is the

most expensive because every request pays for wide-area transmission and for the use of a large accelerator running the full reference model, whose energy cost per request is much higher than that of a compressed variant on an edge node. Savings relative to edge-only execution are smaller, between 4 and 11 percent and about 7 percent on average, because edge-only execution already avoids the network and uses a compressed model; the proposed approach gains from better placement, from selecting cheaper variants for simple requests, and from consolidating load to reduce idle energy. All three curves grow sublinearly: a sixfold increase in tasks raises energy by roughly a factor of four, because fixed costs such as idle power and container start-up are amortized over more tasks. Energy per task at 300 tasks is about 132 joules for the proposed approach, 137 joules for edge execution, and 267 joules for cloud execution.

The narrowing gap between the proposed approach and edge execution at the highest load is worth noting. At 300 tasks the saving over edge execution is only about 4 percent. A plausible reading is that as edge nodes approach saturation, there is less room for consolidation and variant switching, and a growing share of requests must be queued at the edge or redirected, so the advantages of flexible placement shrink.

B. Latency and Service Level Compliance

Table IV reports latency and service level compliance at a load of 300 tasks. Edge-only execution had a median response time of 1.9 seconds but a long tail, with a 95th percentile of 6.8 seconds and 14.2 percent of requests violating their bounds, since saturated nodes build queues. Cloud-only execution had a lower median of 1.4 seconds and a 95th percentile of 3.1 seconds, with a violation rate of 4.1 percent, but it paid for this with the highest energy. The static hybrid policy lay between the two, with 8.7 percent violations. The proposed approach achieved a median of 1.5 seconds, a 95th percentile of 3.4 seconds, and only 2.3 percent violations, the best compliance of all, because predictive provisioning and the service level feedback loop keep nodes away from saturation and send overflow to the cloud before queues build up.

TABLE IV. Latency and Service Level Compliance at 300 Tasks

Policy	Median response (s)	95th percentile (s)	SLA violations (%)	Energy per task (J)
Edge-only	1.9	6.8	14.2	137
Cloud-only	1.4	3.1	4.1	267
Static hybrid	1.6	4.9	8.7	174
Proposed	1.5	3.4	2.3	132

C. Model Variants: Quality and Energy Trade-offs

Table V summarizes the profile of each variant on a representative edge node with an embedded accelerator, except for the reference model, which is measured on the cloud server. Moving from the reference to eight-bit quantization preserves 99.1 percent of quality while reducing memory by roughly half. Four-bit quantization retains 97.3 percent of quality at a memory footprint of 4.4 gigabytes and about 128 joules per request, and is about 26 percent cheaper in energy than the eight-bit version. The pruned and four-bit variant saves a further modest amount of energy at a quality of

95.6 percent. The distilled three-billion-parameter model uses about 71 joules per request and retains 93.8 percent of quality, while the one-billion-parameter variant uses only 29 joules at 88.5 percent. The last variant falls below the 90 percent quality floor of the most demanding request classes, so the decision engine uses it only for classes with relaxed requirements, such as simple classification. The catalogue thus provides a controlled range of operating points, and the energy savings of the proposed approach arise in part from assigning each request class to the cheapest variant that satisfies its quality constraint.

TABLE V. Profile of Model Variants on a Representative Edge Node

Variant	Memory (GB)	Relative quality (%)	Energy per request (J)	Tokens per second
Reference, 16-bit (cloud)	14.0	100.0	266	38
Quantized 8-bit	7.5	99.1	172	8
Quantized 4-bit	4.4	97.3	128	11
Pruned and 4-bit	3.6	95.6	112	13
Distilled 3B, 4-bit	2.1	93.8	71	24
Distilled 1B, 8-bit	1.2	88.5	29	41

D. Effect of Predictive Analysis

The predictive analysis module forecast node utilization with an average error of 6.8 percent, compared with 14.1 percent for a simple persistence forecast that assumes the next interval will resemble the last. With these forecasts, the number of cold container starts that coincided with request arrivals fell from 11.3 per hour to 3.2, and the share of total energy consumed by idle nodes fell from 24 percent to 15 percent because the optimizer could consolidate load and place unneeded nodes in low-power states with confidence.

E. Federated Adaptation Cost and Accuracy

Table VI evaluates the federated loop of Fig. 2 with three data owners holding different local datasets. Adapting only locally, without sharing updates, yields an accuracy of 71.2 percent on a held-out test set that covers all three domains, because each owner sees only part of the distribution. Full-parameter federated training reaches 79.1 percent but requires each client to upload about 14,000 megabytes per round and consumes roughly 46.5 kilojoules of client energy per round, which is impractical on edge links and batteries. Adapter-based federated learning reaches 78.4 percent, within 0.7 points of full-parameter training, while uploading only about 40 megabytes per round, a reduction of more than 99 percent, and using about 4.2 kilojoules of client energy, roughly one eleventh. Adding update compression reduces the upload to about 11 megabytes and client energy to 3.9 kilojoules at a cost of only 0.4 points of accuracy. Centralized training on pooled data, which would not preserve privacy, reached 79.6 percent, so the best federated configuration is within about 1.6 points of the centralized upper bound.

TABLE VI. Federated Adaptation Cost and Accuracy with Three Data Owners

Scheme	Accuracy (%)	Upload per round (MB)	Client energy per round (kJ)	Preserves privacy
Local adaptation only	71.2	0	3.4	Yes
Full-parameter federated	79.1	14,000	46.5	Yes
Adapter-based federated	78.4	40	4.2	Yes
Adapter with compression	78.0	11	3.9	Yes
Centralized, pooled data	79.6	Not applicable	Not applicable	No

F. Ablation Study

Table VII removes one component at a time at a load of 300 tasks. Removing predictive analysis raises energy from about 39,500 to 43,800 joules and doubles service level violations to 4.6 percent, because capacity is prepared late and consolidation becomes conservative. Removing variant selection and fixing the four-bit seven-billion-parameter model raises energy to 47,900 joules, a 21 percent increase, showing that model-level choices contribute as much as placement. Removing the service level agreement manager yields the lowest energy, 36,100 joules, but violations rise to 11.8 percent, showing that part of the energy savings of an unconstrained optimizer is paid for with unacceptable service. Replacing the optimizer with the static hybrid policy yields 52,300 joules and 8.7 percent violations. Each component contributes, and the full framework gives the best balance between energy and service level compliance.

TABLE VII. Ablation of Framework Components at 300 Tasks

Configuration	Total energy (J)	SLA violations (%)	Change in energy (%)
Full framework	39,500	2.3	0
Without predictive analysis	43,800	4.6	+11
Without variant selection (fixed 4-bit 7B)	47,900	2.9	+21
Without SLA manager	36,100	11.8	-9
Without energy optimization (static hybrid)	52,300	8.7	+32

DISCUSSION

The results support the central proposition that energy efficiency for LLMs at the edge should be pursued jointly at the model level and the system level. Fig. 3 and Table III show that executing everything in the cloud costs roughly twice as much energy as the alternatives across all loads, which confirms that wide-area transfer and large-accelerator execution of the reference model are expensive for workloads that can be served well by compressed variants nearby. At the same time, edge-only execution is not a complete answer: its tail latency and violation rate in Table IV show that small nodes saturate under bursts, and an approach that cannot overflow to the cloud will break service levels. The proposed framework combines the strengths of both by placing most requests at the edge with the cheapest adequate variant and using the cloud selectively.

Three aspects of the design deserve emphasis. First, the variant catalogue

converts quality into a controllable resource. The decision engine can spend quality headroom where it is not needed, as for simple classification, and protect it where it is, as for summarization. Second, prediction matters not only for performance but for energy, because idle power is a large share of the consumption of lightly loaded edge nodes, and consolidation is safe only if demand can be anticipated. Third, the service level manager acts as a counterweight to the optimizer. The ablation shows that without it, energy appears to improve while service quality deteriorates sharply, a reminder that energy results should always be reported alongside service level compliance.

The federated loop in Fig. 2 broadens the framework from inference to adaptation. Adapters reduce uploads by more than two orders of magnitude relative to full updates, which turns federated adaptation from impractical to routine on edge networks, with a small loss of accuracy.

Several limitations should be noted. The experiments were conducted on a modest testbed with a limited number of node types, and results on larger and more varied deployments may differ. The energy comparison in Fig. 3 reports totals for a single workload family, and the cloud measurements include an allowance for host overhead that is an estimate rather than a metered value; the carbon impact depends additionally on the energy mix at each site, which was not modeled. The quality measures are task-specific scores relative to a reference model and do not capture all aspects of language model behavior such as safety, factuality in rare cases, or bias, which may change under compression. The heuristic optimizer offers no optimality guarantee, and its behavior under adversarial or highly unpredictable loads was not studied. Federated updates can leak information about local data, and the framework does not include formal privacy guarantees.

Conclusions

This paper presented an energy-aware framework for running large language models at the edge. The framework combines a container-based offloading layer with a service level agreement manager, a virtual machine manager, a directory server, acceleration servers, and monitoring, an energy efficiency decision engine with predictive analysis and energy optimization, a catalogue of compressed model variants, and a federated adaptation loop based on lightweight adapters. Decisions about where to run each request, which variant to use, and how to batch are made jointly to minimize predicted energy under latency and quality constraints.

Across task loads from 50 to 300, the proposed approach used the least energy at every load, about 53 percent less than cloud-only execution and about 7 percent less than edge-only execution on average, while keeping service level violations at 2.3 percent compared with 14.2 percent for edge-only execution. Four-bit quantization preserved 97.3 percent of the reference quality at about half the energy of the eight-bit variant, predictive analysis halved cold starts and reduced idle energy, and adapter-based federated learning cut client upload by more than 99 percent at a small cost in accuracy. These findings indicate that energy-efficient LLM deployment at the edge is best treated as a joint optimization of models, placement, prediction, and service levels, rather than as a problem of compression or offloading alone.

REFERENCES

[1] Pawan Kumar Singh, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5713

[2] Pawan Kumar Singh, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5712

[3] Rangavinay Teja Royal Jagga, 2026, DOI: 10.1109/GCWCN66157.2025.11448495, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,https://ieeexplore.ieee.org/document/11448495

[4] Rangavinay Teja Royal Jagga , 2026, Subtle Data Contamination in Visual AI: Covert Model Backdoors via Resampling Exploits, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,https://ieeexplore.ieee.org/document/11448514

[5] Rangavinay Teja Royal Jagga, 2026, Robotic Tactile-Vision Integration for Interactive Object Disentanglement, https://ieeexplore.ieee.org/document/11448337

[6] Rangavinay Teja Royal Jagga, 2026, A Discounted Future Reward Framework for Intelligent Loyalty Optimization using CLTV-Aware Segmentation and Churn Prediction, https://ieeexplore.ieee.org/document/11497560

[7] Anjaneyulu Gudla; Manish Kumar; Vamshi Jeksani; Kiran Kumar Reddy; Gayathri Band, Harnessing Artificial Intelligence Cybersecurity: Cutting Edge Collaboration of SAP AWS to Safeguard Life Science Data. 2026 https://doi.org/10.1109/aiei69164.2026.11497833

[8] G. Mahender, Department of CSE(Data Science), Vardhaman College of Engineering ; Vamshi Jeksani; Manish Kumar; Koppuravuri Sri Lakshmi Sruthi; Kiran Kumar Reddy, SAP Cloud System on AWS for Risk Detection and Integration Artificial Intelligence Scalable Cybersecurity 2026 https://doi.org/10.1109/aiei69164.2026.11497435

[9] Kiran Kumar Reddy; Kamalakar Gunupati; Manish Kumar; Pell Reddy Rajender Reddy; Ramesh Julakanti; Ram Reddy Jonnalagadda, SAP System Optimization Using AI-Driven Process Automation and Predictive Modeling Maintenance for Enhanced Business Efficiency. 2025 https://doi.org/10.1109/computingcon64838.2025.11376613

[10] Amilineni, M.V. 2025. Audit-Safe Agentic RAG Framework for Regulated Enterprise Systems: A Trustworthy AI Architecture for SAP, Finance, Healthcare, and Government Workflows. INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES. (Jan. 2025), 34–46. DOI: https://doi.org/10.29284/7fjzk883

[11] Pawan Kumar Singh, ACCELERATING LARGE LANGUAGE MODEL TRAINING USING HYBRID QUANTUM-CLASSICAL VARIATIONAL AUTOENCODERS WITH INTEGRATED PHOTONIC QRAM, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/501, DOI: https://doi.org/10.46121/pspc.54.3.45

[12] Pawan Kumar Singh , RESILIENCE ASSESSMENT OF MODERN POWER GRID INFRASTRUCTURE AGAINST EXTREME WEATHER USING PHYSICS-INFORMED NEURAL NETWORKS, Vol. 54 No. 1 (2026):

January-March 2026, Power System Protection and Control, ISSN-1674-3415,https://pspac.info/index.php/dlbh/article/view/502, DOI: https://doi.org/10.46121/pspc.54.1.64

[13] Pawan Kumar Singh, MITIGATING THE EXABYTE STORAGE CRISIS: ENHANCING READ/WRITE EFFICIENCY IN 5D OPTICAL GLASS STORAGE USING SILVER NANOPARTICLE DOPING, Vol. 38 No. 11s (2025), International Journal of Applied Mathematics (IJAM) , pISSN 1311-1728, eISSN 1314-8060 ,https://ijamjournal.org/ijam/publication/index.php/ijam/article/view/2153

[14] Mr. Pawan Kumar Singh (Author), Miss. Pari Nidhi Singh (Author) ,Intelligent Bot, ASIN : B0DBGCQY8X,Publisher : Namya Press, Publication date : November 25, 2022,Language : English, Print length : 105 pages,ISBN-10 : 9390124867,ISBN-13 : 978-9390124862,Item Weight : 7.5 ounces, Dimensions : 6.24 x 0.43 x 9.24 inches,https://www.amazon.com/Intelligent-Bot-Pawan-Kumar-Singh/dp/B0DBGCQY8X/ref=sr_1_5?crid=2VAXN5BRN5SMS&dib=eyJ2IjoiMSJ9.v3OCQvHBpy20RIjfMWhoqbOl-RLOlUJIC8Xa0Pjeqilvo2sbZ_suN3LbSCw_OA_ZD_4lgjRLMGlBwZQQ1ehZEfDpt_MJiOUfVuqqzMav0I5_mb6pN5DQog0U7SA826n1DQQ3QAtWwopHy4lC5IfhZXttKj9faiyATwhdtDATSSV1-r7FmzM2VQyIA0jxcfq6E6hp9ECfH1fby0li9kUfGpLxMq4GktlzUIJO5ggrhsQ.G1AjQiVqU4HFsnSczvz-X6NXCNs3EFSn-RI8mcWWGOc&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book&qid=1787815741&sprefix=pawan+kumar+singh%2C+bo%2Caps%2C450&sr=8-5

[15] Mr. Pawan Kumar Singh (Author), STORAGE CRISES DUE TO RISE OF AI: A Comprehensive Guide to Managing the Storage Challenges in the Age of Artificial Intelligence, ASIN : B0HFK12ZCT
Publisher : EB1 Profile, Accessibility : Learn more, Publication date : August 17, 2026,Edition : 1st
Language : English, File size : 1.7 MB, Screen Reader : Supported, Enhanced typesetting : Enabled
X-Ray : Not Enabled, Word Wise : Not Enabled ,Print length : 261 pages ,ISBN-13 : 979-8951479112,Page Flip : Enabled, https://www.amazon.com/STORAGE-CRISES-DUE-RISE-Comprehensive-ebook/dp/B0HFK12ZCT/ref=sr_1_1?crid=9MF6C4WDVJR5&dib=eyJ2IjoiMSJ9.DQ8fesBch7ELmRnilDt-YA.NBSCHLmetvnbhkxwIpOoj3s0IVsCV8y6-ZCfCP6Ul1c&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book%2C+eb1+Profile&qid=1787816040&sprefix=pawan+kumar+singh%2C+book%2C+eb1+profile%2Caps%2C420&sr=8-1

[16] Parag Bharatkumar Parikh, FRAMEWORKS FOR MITIGATING SUPPLY CHAIN DISRUPTIONS AND MATERIAL PRICE VOLATILITY IN SUSTAINABLE BUILDING CONSTRUCTION, Vol. 53 No. 2 (2025): April-June 2025 , , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/447, DOI: https://doi.org/10.46121/pspc.53.2.34

[17] Parag Bharatkumar Parikh, IMPLEMENTING DIGITAL INSPECTION PROTOCOLS FOR ENHANCED QUALITY CONTROL MANAGEMENT IN HIGH-RISE CONSTRUCTION, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/439, DOI: https://doi.org/10.46121/pspc.54.3.12

[18] Parag Bharatkumar Parikh , EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Power System Protection and

Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/issue/view/30, DOI: https://doi.org/10.46121/pspc.54.1.61
[19] Parag Bharatkumar Parikh, EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Vol. 54 No. 1 (2026): January-March 2026, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/440, DOI: https://doi.org/10.46121/pspc.54.1.61
[20] Ronakkumar shah, REDEFINING BUSINESS PROCESS MANAGEMENT: THE SYNERGY OF AI AND INTELLIGENT AUTOMATION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/448, DOI: https://doi.org/10.46121/pspc.54.3.14
[21] Ronakkumar shah, OPERATIONAL ACCOUNTABILITY IN ACTION: METRICS AND MECHANISMS FOR SYSTEMIC SUCCESS, Vol. 53 No. 4 (2025): October-December 2025, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/452, DOI: https://doi.org/10.46121/pspc.53.4.40
[22] Ronakkumar shah, BEYOND THE BLUEPRINT: AUTOMATED WORKFLOW ENFORCEMENT AND POLICY GOVERNANCE, Vol. 53 No. 2 (2025): April-June 2025 , System Protection and Control, ISSN-1674-3415 ,https://pspac.info/index.php/dlbh/article/view/451, DOI: https://doi.org/10.46121/pspc.53.2.35
[23] Tushita Agarwal; Swaminathan Sethuraman, Mapping Customer Behavior: Visual Analysis of Online Retail Interactions, Date Added to IEEE Xplore: 17 February 2026, DOI: https://doi.org/10.1109/IC2NC67409.2025.11376261
[24] Tushita Agarwal, Efficient Data Summarization and Comparison via Grid-Based Statistical Proxies for Large-Scale Datasets, Date Added to IEEE Xplore: 03 December 2025, DOI: https://doi.org/10.1109/ICBATS66542.2025.11258564
[25] Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021,DOI: https://doi.org/10.30574/ijsra.2021.2.1.0047
[26] Dip Bharatbhai Patel, University of North America, Virginia, USA, Home / Archives / Vol. 5 No. 02 (2025): Volume 05 Issue 02 / Articles, https://www.academicpublishers.org/journals/index.php/ijdsml/article/view/5958
[27] Sai Akshara Vemuri , KINEMATIC OPTIMISATION OF SOFT ROBOTIC MANIPULATORS FOR MEDICAL INTERVENTION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/450
[28] Sai Akshara Vemuri, https://www.periodicos.ulbra.org/index.php/acta/article/view/528
[29] Aziz, A., Chandusha, K., Naga Kiran, B., Prasad, M. L. M., Golla, K., Rajeswari, S., & Katta, S. K. R. (2026). Exploring Machine Learning Applications for Genomic Data Analysis in Personalized Medicine. International Journal of Drug Delivery Technology, 16(38s), 1166-1173. https://doi.org/10.25258/ijddt.16.38s.127,https://www.periodicos.ulbra.org/index.php/acta/article/view/687
[30] Mohammad Naffizuddin , Chaganti Ashok ,Sowjanya Gunukula , Rvbs Sarma , Bhavana Sujanamulk , Bharani Krishna Takkella , Chukka Ram Sunil ,Varri Sujana , Ashwini Kumar, Etiological Factors of Orofacial Clefts in a Hospital-Based Population: A Descriptive Analytical Study, Published: December 10, 2025,DOI: 10.7759/cureus.98868,DOI : https://doi.org/10.7759/cureus.98868

[31] Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep), Publication year 2021, Volume 2021, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/970
[32] Manoj Kumar Kagitha, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/969
[33] Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS),Publication year 2023, Volume 2023, Issue 12,https://computerfraudsecurity.com/index.php/journal/article/view/971
[34] Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, https://www.eudoxuspress.com/index.php/pub/article/view/5054
[35] Lakshmi Rahul Reddy Mareddy. (2026). Security and Performance Assessment of Encryption Techniques for Cloud Platforms. International Journal of Communication Networks and Information Security (IJCNIS), 18(1), 68–80. Retrieved from https://ijcnis.org/index.php/ijcnis/article/view/8735
[36] Lakshmi RahulReddy Mareddy, Context-Aware Learning Approaches for Improving Prediction Accuracy in Dynamic Systems, International Journal of Computational Mathematical Ideas (IJCMI), Vol.17 No.3(2025): IJCMI Vol17 NO3 2025 https://www.ijcmi.com/index.php/journal/article/view/15
[37] Lakshmi Rahul Reddy , Mareddy. (2026). Revisiting Autonomy in Artificial Intelligence through Agentic System Design. Journal of Computational Analysis and Applications (JoCAAA), 35(1), 707–713. https://eudoxuspress.com/index.php/pub/article/view/4780
[38] Mahendrakumar Kalal, Lean-Driven SAP Production Planning Optimization Using Machine Learning for Inventory and Throughput Efficiency, Date of Conference: 19-20 March 2026, Date Added to IEEE Xplore: 08 April 2026,ISBN Information:,ISSN Information: DOI:10.1109/ISDFS69419.2026.11459029, https://ieeexplore.ieee.org/abstract/document/11459029
[39] Mahendra kumar Kalal, SUPPLY CHAIN RESILIENCE THROUGH INTEGRATED SAP PRODUCTION PLANNING – PROCUREMENT SYNCHRONIZATION, Archives / Vol. 54 No. 1 (2026): January-March 2026 / Articles, https://pspac.info/index.php/dlbh/article/view/259
[40] Dr Manoj Goel (December 2018). THE ATTITUDE OF DIFFERENT STATES STUDENTS TOWARDS PHYSICAL EDUCATION; STUDYING IN PHYSICAL EDUCATION COLLEGES OF MAHARSHTRA STATE, https://ijeponline.com/index.php/journal/article/view/965/847
[41] Ilakiya Ulaganathan, Tagore Engg. College, Anna University, Chennai, India, ; FNU Antara; Santhosh Chitraju Gopal Varma; Zokir Mamadiyarov; Ziyadulla Turayev, Published in: 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Date of Conference: 22-23 August 2025, Date Added to IEEE Xplore: 19 December 2025, ISBN Information: https://ieeexplore.ieee.org/document/11294748/references#references
[42] Shreyas Kasture, Sudhanshu Maurya, Amit Kumar Sharma, Santhosh Chitraju Gopal Varma, Kashish Mirza, Firdous Sadaf Mohammad Ismail, DOI: http://doi.org/10.11591/ijai.v15.i2.pp1746-1761, IAES International Journal of Artificial Intelligence (IJ-AI), ISSN/e-ISSN 2089-4872/2252-8938 https://ijai.iaescore.com/index.php/IJAI/article/view/26932
[43] Shiv Shankar Dwivedi, Mr. Santosh Kumar Sharma, Future-Proofing Data Security with Quantum-Resistant Cryptography, Vol. 11 No. 3 (2025):

IJCESEN, DOI: https://doi.org/10.22399/ijcesen.3447, https://ijcesen.com/index.php/ijcesen/article/view/3447
[44] Vijender Kalmotia, Enhancing statistical accuracy in gene perturbation studies, Volume 266, August 2026, 105819, https://doi.org/10.1016/j.biosystems.2026.105819, https://www.sciencedirect.com/science/article/pii/S0303264726001292
[45] Naveen Undrathi, Design and implementation of generative AI-driven data transformation pipelines using spring boot for scalable financial applications, eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || CrossRef DOI, September 2026 (Volume 31, Issue 3), DOI url: https://doi.org/10.30574/wjarr.2026.30.3.1629,
[46] Naveen Undrathi, Agentic generative AI-enabled microservices framework for intelligent financial data processing in java-based enterprise systems, September 2026 (Volume 31, Issue 3), eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || https://doi.org/10.30574/wjarr.2026.30.3.1628
[47] Akshay Jain, Comparative Analysis of Linkage Methods in Hierarchical Clustering for Customer Segmentation, Conference paper, First Online: 07 October 2025, https://link.springer.com/chapter/10.1007/978-981-96-7289-9_44
[48] Ismail Hossain, Department of Computer Science, George Mason University, Fairfax, VA, USA, Mridul Banik, Department of Computer Science, Colorado State University, Fort Collins, CO, USA, https://ieeexplore.ieee.org/abstract/document/11446534 ,,DOI: 10.1109/FMLDS67896.2025.00106
[49] Ismail Hossain, Mridul Banik , Fahmid Al Farid , Jia Uddin , Hezerul bin Abdul Karim , Advancing Large Language Models for Low-Resource Languages: A Systematic Review of Pretraining, Adaptation, and Ethical Challenges, 10.32604/cmes.2026.075507, Date: 2026 https://www.sciencedirect.com/org/science/article/pii/S152614922600322X
[50] Ismail Hossain; Mridul Banik, Bridging Visual and Linguistic Intelligence for Chest X-rays: A Comprehensive Review of ViTs and LLM Synergies, Published in: 2025 IEEE International Conference on Future Machine Learning and Data Science (FMLDS) Date of Conference: 02-05 November 2025, Date Added to IEEE Xplore: 25 March 2026, ISBN Information: https://ieeexplore.ieee.org/abstract/document/11446534
[51] Rifat Sarker Aoyon1, Ismail Hossain2, M. Abdullah-Al-Wadud3, Jia Uddin4, A Secured and Continuously Developing Methodology for Breast Cancer Image Segmentation via U-Net Based Architecture and Distributed Data Training, https://www.sciopen.com/article/10.32604/cmes.2025.060917
[52] M. B. Vibha, M. Chandrika, Samitha Khaiyum & P. Rakshitha Kiran, Predicting sedentary behavior in adults using stacked LSTM modeling, published: 27 November 2024
Volume 16, pages 346–355 (2025), https://link.springer.com/article/10.1007/s13198-024-02622-2
[53] Bellary Chiterki Anil, Jayasimha Sondekoppa Rajkumar, T. L. Divya, Samitha khaiyum
Rakshitha Kiran P., Balakrishnan Ramadoss, A Radiomics-based Framework for Liver Cancer Analysis using Explainable Artificial Intelligence (XAI) Methods, Vol. 15 No. 3 (2025): June, 2025, https://etasr.com/index.php/ETASR/article/view/10377
[54] Jayminkumar Patel, Sanjaykumar patel, Birju Patel, Rajan shah, Development and Validation of a Stability-Indicating Analytical Method for Quantitative Determination of Selected Anticancer Drugs in Pharmaceutical Dosage Form, Archives / Vol. 35 No. S2 (2026): Apr-jun 2026, https://welltestingjournal.com/index.php/WT/article/view/288
[55] Jayminkumar Patel, Sanjaykumar patel, Birju Patel, Rajan shah, Artificial

Intelligence-Guided Optimization of Chromatographic Methods for Pharmaceutical Analytical Development, Archives / Vol. 35 No. S2 (2026): Apr-jun 2026, https://welltestingjournal.com/index.php/WT/article/view/294

[56] Sharoon Sharif, Inikpi Egbunu, Yujian Fu, and Zhigang Xiao. 2026. Hybrid Multi-Modal Learning for Extreme Long-Tail Fungi Classification. In 2026 ACM Southeast Conference (ACMSE 2026), April 23–25, 2026, Troy, AL, USA. ACM, New York, NY, USA, 38–45. https://doi.org/10.1145/3746467.3801502

[57] A. Chandrakumaran, "Innovative Strategies for Real-Time Analytics in Modern Data Warehousing Using Dockerized Ecosystems," 2025 International Conference on Advances in Next-Gen Computer Science (ICANCS), Bangalore, India, 2025, pp. 1-6, doi: https://doi.org/10.1109/ICANCS65819.2025.11377462

[58] A. Chandrakumaran, "Enhancing SAP Predictive Maintenance and Industrial Analytics with LLM-Driven Automation," 2025 IEEE Madhya Pradesh Section Conference (MPCON), Jabalpur, India, 2025, pp. 238-244, doi: https://doi.org/10.1109/MPCON66082.2025.11256577.

[59] A. Chandrakumaran, "Enhancing SAP Cloud Optimization with Explainable AI: Cost Control, Workload Forecasting, and Performance Enhancement," 2025 3rd International Conference on Inventive Computing and Informatics (ICICI), Bangalore, India, 2025, pp. 456-463, doi: https://doi.org/10.1109/ICICI65870.2025.11069968.

[60] A. Chandrakumaran, "Innovative Strategies for Real-Time Analytics in Modern Data Warehousing Using Dockerized Ecosystems," 2025 International Conference on Advances in Next-Gen Computer Science (ICANCS), Bangalore, India, 2025, pp. 1-6, doi: https://doi.org/10.1109/ICANCS65819.2025.11377462.

[61] Ashwini Chandrakumara, year :2025, Leveraging Autonomous AI for Agile Growth: Transforming Project Management in Fast-Scaling Startups, Proceedings of the 2nd International Conference on Sustainable Business Practices and Innovative Models (ICSBPIM-2025), pages :730-743, issn :2352-5428, isbn :978-94-6463-872-1, Atlantis Press https://doi.org/10.2991/978-94-6463-872-1_46

[62] Anasuya Patil 1, Gurinderdeep Singh 2, Rajendra Dnyandeo Dighe 3, Dhruv Dev 4, Bhaveshkumar A Patel 5, Samatha Rudrangi 6, Gaurav Tiwari 7, Preparation, optimization, and evaluation of ligand-tethered atovaquone-proguanil-loaded nanoparticles for malaria treatment, 2025 Apr;36(6):711-742. doi: 10.1080/09205063.2024.2422704. Epub 2024 Nov 10., https://pubmed.ncbi.nlm.nih.gov/39522102/

[63] Kancharla Vijayalakshmi, Bethapudi Samuel Anand Andrews, Estimation of Cyproheptadine Hydrochloride and Tricholine Citrate Simultaneously in Syrup Dose by Applying Stability Indicating HPLC Methodology, December 31, 2020, Vol. 11, Issue 4, Pages 1813–1819, DOI: 10.26452/ijrps.v11iSPL4.4383, https://ijrps.com/v11si4/3683

[64] Vijayalakshmi Kancharla ,Kumar Abbaraju V D N, Andrews B S A, Nagaraju Dasari, Nagu Korupolu, Melphan as Anticancer and New Insights of Drug as Parenteral Dosage form against SARS-CoV-2 by Using Development and Validation , November 18, 2020, Vol. 11, Issue 1, Pages 1437–1446 DOI: 10.26452/ijrps.v11iSPL1.3686, https://ijrps.com/v11si1/2911

[65] Ramesh, D., A. Singh, and L. H. Channaiah. 2026. “ Evaluating the Effectiveness of Chewy Chocolate Chip Granola Bar Manufacturing Process on Salmonella Survival.” Food Science & Nutrition 14, no. 4: e71698. https://doi.org/10.1002/fsn3.71698.

[66] Inventor: Akhilesh RaiDilip Srinivas Krovi Sarthak Saptarshi Bandopadhyay Ravinder Singh Choudhary Richard LAWTON, 2025-02-28, Priority to

US19/067,141,2025-11-14,Assigned to CITIBANK, N.A.,2025-12-30,Application granted,2025-12-30, Publication of US12511187B, https://patents.google.com/patent/US12511187B1/en?oq=US12511187

[67] Vivekanandan Govindan Ekambaram , REAL-TIME AUTOMATED DEFECT DETECTION IN INDUSTRIAL MANUFACTURING USING DEEP LEARNING AND TEXTURE ANALYSIS, Archives ,Vol. 54 No. 4 (2026): October-December 2026, Power System Protection and Control, ISSN-1674-3415,DOI: https://doi.org/10.46121/pspc.54.4.06, https://pspac.info/index.php/dlbh/article/view/545

[68] Vivekanandan Govindan Ekambaram, Yamuna Natarajan, AN OPTIMIZED HYBRID FRAMEWORK FOR REAL-TIME EDGE DETECTION AND FEATURE EXTRACTION IN AUTONOMOUS VEHICLE VISION SYSTEMS, Vol. 54 No. 4 (2026): October-December 2026 , Power System Protection and Control, ISSN-1674-3415, DOI: https://doi.org/10.46121/pspc.54.4.05, https://pspac.info/index.php/dlbh/article/view/544

[69] John Smart Mgbakor, Behavioral Health Equity In High-Disparity Urban Communities: Public Health Impact, National Scalability, And Clinical Practice Implications, Pages1166 – 1177Cover Date, August 2026, Volume34, Issue, Special Issue 7- Fundamental Studies, Print ISSN-0965-7452, Electronic- ISSN-2396-8893, DOI: 10.1922/ejprd.v34i7s.1767, https://www.ejprd.org/article.php?article_id=1767&journal_id=171

[70] Nikhil Kumar Singh, Binay K. Singh, Piyush Kumar, Avinash Pandey, Sudhir Kumar, Sujit Kumar Bishi, A. Pattanayak, V. P. Bhadana, Sujay Rakshit & Kishor U. Tribhuvan, Integrative genomic and machine learning approaches reveal evolutionary signatures in the winged bean mitochondrial genome, Published: 28 November 2025,Volume 26, article number 3 (2026), https://link.springer.com/article/10.1186/s12870-025-07738-6

[71] GAURAV KUMAR SINHA, Year 2026 https://mathsapplication.com/article-view/?id=223

[72] GAURAV KUMAR SINHA, Year 2026 , https://mathsapplication.com/article-view/?id=224

[73] GAURAV KUMAR SINHA, Year 2026 , https://mathsapplication.com/article-view/?id=225

[74] Rasmi Ranjan Choudhury, Design and Implementation of a Tuned Rear Axle Bending Damper for Beat Boom Noise Mitigation in SUV Driveline Systems, 2026 International Conference, https://ieeexplore.ieee.org/document/11746870/authors#authors

[75] Rasmi Ranjan Choudhury, Comparative Performance Analysis of Single-Speed Versus Multi-Speed Transmissions in Battery Electric, 2026 International Conference, https://ieeexplore.ieee.org/document/11746704

[76] Nithesh Gudipuri1, Lalith Chandra Bandaru2, Mohammed Shakeer Bandrevu, SCALABLE DATA ENGINEERING FRAMEWORKS FOR REAL-TIME PROCESSING IN DISTRIBUTED SYSTEMS, Published:2026-10-18,Pages:444-453, https://actainnovations.com/index.php/pub/article/view/1900 ,doi: 10.62441/actainnovations.vi.1900

[77] Mohammed Shakeer Bandrevu; Pratham Patkar; Janardhana Naidu Kola, Trustworthy AI Models for Secure Data Processing Environments, Published in: 2026 International Conference on Natural Language Processing and Computer Vision (ICNPCV),Date of Conference: 17-18 July 2026, DOI: 10.1109/ICNPCV69434.2026.11708997, https://ieeexplore.ieee.org/document/11708997/authors#authors,

[78] Lalith Chandra Bandaru; Venkateswara Rao Batta; Uday Kumar Reddy Gangula, Hybrid AI and Data Analytics Frameworks for Cyber Resilience, 2026 International Conference on Natural Language Processing and Computer Vision (ICNPCV),Date of Conference: 17-18 July 2026,DOI: 10.1109/ICNPCV69434.2026.11708795,

https://ieeexplore.ieee.org/document/11708795
[79] Lalith Chandra Bandaru; Karimulla Syed,Head of Access Management, S&P Global, Raleigh, NC, USA
[80] ; Nishanth Vepachedu; Rajmani Singh Oinam; Raja Vignesh Vaithiyalingam; Quazi Taif Sadat, Advanced Secure Knowledge Extraction Techniques for Large Language Models, 2026 International Conference on Emerging in Computing Technologies (INCECT),Date of Conference: 21-22 August 2026, DOI: 10.1109/INCECT72370.2026.11742788, https://ieeexplore.ieee.org/document/11742788
[81] Mohammed Shakeer Bandrevu, Independent Researcher, Aubrey, TX, USA, ; NagaSatyanarayana Raju Uppalapati; Ramesh Bellamkonda; Srikanth Kamatala, Llm-Augmented Cyber Threat Detection Using Federated Edge Intelligence and Data Mining Techniques, Published in: 2025 International Conference on Computing Technologies & Data Communication (ICCTDC),Date of Conference: 04-05 July 2025, https://ieeexplore.ieee.org/document/11158153 , DOI: 10.1109/ICCTDC64446.2025.11158153
[82] Ankit Pathak, Leveraging AI for Better Data Quality and Insights, Volume 7 Issue 3 (2025), https://doi.org/10.32996/jcsts.2025.7.2.19, https://al-kindipublisher.com/index.php/jcsts/article/view/9345
[83] Uma Rajeswari Batchu, Bharati Reddi, Joshna Rani Surapaneni, Aruna Jangam, Sunil Misra, Anthony Addlagatta & Prakasham Reddy Shetty, Inhibition of Xanthine Oxidase by 1-O-Methyl Chrysophanol, A Hydroxyanthraquinone Isolated from Amycolatopsis Thermoflava ICTA 103, Published: 13 December 2025,Volume 66, pages 1049–1068 (2026) https://link.springer.com/article/10.1007/s12088-025-01534-4
[84] GAURAV KUMAR SINHA, Year 2025, https://visualcompublications.es/SAUC/article/view/6525
[85] GAURAV KUMAR SINHA, Year 2025, https://visualcompublications.es/SAUC/article/view/6526
[86] GAURAV KUMAR SINHA, Year 2025, https://visualcompublications.es/SAUC/article/view/6527
[87] D Benito Johnson 1, Gurinderdeep Singh 2, Deeksha Sharma 3, Venkatesan Natarajan 4, Knv Chenchu Lakshmi 5, Ram C Dhakar 6, Sadhana R Shahi 7, Suresh Velayutham 8, Ruchi Tiwari 9, Exploring Computational Advancements in ADME: Essential Insights for Drug Disposition, PMID: 39496312 DOI: 10.62958/j.cjap.2024.033, https://pubmed.ncbi.nlm.nih.gov/39496312/
[88] Mohammed Shakeer Bandrevu, Service Now Ebonding Methods, Implementation and Key features Published,30 Nov 2024, https://ijcttjournal.org/archives/ijctt-v72i11p125