International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 6, Issue 3, 160-169, 2025
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V6I3P121

Original Article

Carbon-Aware Dynamic Batching for Deep Learning Inference: Optimizing the Energy-Latency Trade-off in High-Frequency Transaction Monitoring

Anvesh Katipelly[1], Sumith Thalary[2]
[1]Senior Software Engineer, PayPal, Texas, USA.
[2]Sr DevOPs Engineer, Kubota Tractor Corp, Dallas, TX, USA.

Received On: 08/07/2025 ***Revised On: 22/07/2025*** ***Accepted On: 16/08/2025*** ***Published On: 07/09/2025***

Abstract - *The increasing demand for real-time analytics in high-frequency transaction monitoring systems has led to a rapid growth in deep learning inference workloads, resulting in significant energy consumption and environmental impact. The current paper manages to introduce a carbon-conscious dynamic batching system that makes sure to trade off energy efficiency and latency in latency-sensitive inferences. The suggested method combines the real-time carbon intensity indicators, the workload characteristics and the performance metrics of the system into one decision-making pipeline. The control mechanism is a feedback-based mechanism dynamically varying the batch sizes according to the transaction arrival rates, queue conditions and service-level agreement (SLA) constraints, to make sure that the latency requirements are always satisfied. At the same time, the framework offers congruence between the execution of inferences and the low-carbon energy intervals, minimizing the carbon footprint. Experimental findings indicate that the proposed system can reduce energy usage and carbon emissions by up to 35% and 40%, respectively, over the conventional methods of performing static batching and still undergoes p95 latency within the strict operational limits. Moreover, the framework has a high scalability and flexibility in different workload conditions such as bursty and high-throughput conditions. With the well-balanced performance and sustainability goals, the work helps to improve the green AI practices and offers a viable solution on the implementation of environmentally friendly deep learning systems in the real-time financial monitoring systems.*

Keywords - *Dynamic Batching, Deep Learning Inference, Energy Efficiency, High-Frequency Transactions, Green AI.*

1. Introduction

The adoption of high-frequency transaction systems in areas of financial services, fraud detection, and real-time monitoring has necessitated the requirement of scalable and low-latency deep learning inference systems. Distributed cloud and smart data pipeline architectures are becoming a common feature of modern data-intensive applications that need to handle huge volumes of transactions as quickly as possible. The relevance of the balance between big data engineering and cloud-native systems in real-time analytics and decision-making in these settings has already been emphasized in the previous works [1,2] (But with increasing loads of inferences, there is an increase in the associated energy consumption, which brings up sustainability and carbon implications in large-scale AI implementations. Latency-aware scheduling and resource optimization are recent developments that underline the importance of the need to meet the performance requirements and the efficiency of the system, especially in networks with mission-critical and public safety needs [3]. Simultaneously, AI-based fraud detection applications and smart wireless systems have shown that there is a necessity to have ultra-low latency and high throughput in financial transactions monitoring [4,5]. Such systems may be tightly provisioned by service level agreements and thus it is difficult to add energy conscious optimizations without compromising performance.

The growing studies in large-scale AI platforms and decision-oriented architectures add more emphasis to the urgent need of adaptive and data-informed optimization approaches to large-scale distributed systems [6-7]. Furthermore, the combination of AI and radio access networks as well as distributed analytics processes indicates that controlling computation, latency, and energy efficiency jointly is becoming increasingly complicated [8] [9]. Here carbon-conscious dynamic batching offers a way forward in order to optimize the energy-latency trade-off in deep learning inference systems. Such frameworks have the potential to support sustainable AI functions by introducing dynamic energy metrics and dynamic control controls, as well as meet high-latency demands of high-frequency transaction systems.

2. Background And Preliminaries

2.1. Deep Learning Inference Pipelines

The intelligent systems built on deep learning inference pipelines are the core of the real-time intelligent systems and

especially in the high-frequency transactions monitoring domains. These pipelines can generally be split into data ingestion, preprocessing, model execution and post processing phases and these stages are subject to stringent latency constraints. The incoming transaction data is first made normal and converted to model-compatible forms and then fed into a trained neural network to provide predictions. The inference engine then produces outputs, e.g. fraud scores or classifications that are further used to make downstream decisions. These pipelines in large-scale deployments are frequently distributed throughout cloud or edge infrastructure in order to provide the ability to scale up and fault tolerance. The coordination of compute, memory and network resources is highly important with any delays even being minimal, that can affect system responsiveness and business results.

2.2. Batching Techniques in Inference Systems

The most popular optimization method in inference systems is known as Batches and it involves combining multiple input requests into one processing unit to enhance the utilization of hardware and throughput. With static batching, batch sizes are set and fixed irrespective of changes in the workloads. Although this method is easily implemented and may result in stable operation give homogeneous loads, such operation may be very inefficient under changing request rates, resulting in either excessive utilization or higher latency.

Conversely, dynamic batching automatically scales the batch size dynamically on the basis of the rates of incoming requests, system load and latency constraints. This strategy allows the utilization of resources better and throughput improvement without compromising response times. Dynamic batching however adds more complexity to the scheduling and control systems, since the system has to keep balancing conflicting aims like latency, throughput, and resource efficiency. In high-frequency transaction systems, dynamic batching is particularly advantageous due to the bursty and unpredictable nature of workloads.

2.3. Energy Consumption in AI Systems

Energy consumption in AI systems has become a critical concern due to the increasing scale of deep learning models and the continuous nature of inference workloads. Although easier to compute than training, inference can use a lot of energy in the deployed context, in particular in real-time systems with millions of transactions to candidates. The main activities contributing to the energy consumption are the use of the GPU/CPU, the access to the memory and the movement of data between the distributed components. Westernization of resource allocation and non-utilization of hardware further increases wastage of energy. With organizations increasing their efforts to achieve a sustainable organization, interest in optimization of inference pipelines has increased to minimize energy usage without quality performance compromised. Workload consolidation, hardware acceleration, and adaptive scheduling are some of the techniques that are being explored to attain energy-efficient AI operations.

2.4. Carbon Intensity Models and Data Sources

Carbon intensity is the ratio of carbon emissions per unit of electricity consumed and this varies widely over time and location depending on the source of energy generation. By incorporating the concept of carbon intensity awareness into AI systems, it is possible to make decisions as being more sustainable by matching the computational workloads with the periods or areas with the least carbon emissions. Carbon intensity models use real time and past information on the energy grids, the combination of renewable and non-renewable energy and forecast on the environmental impact. The typical sources of data are the national grid operators, energy monitoring platforms and APIs which offer real-time carbon measurements. The systems can use these signals to dynamically optimize operations to reduce carbon footprint without impacting service-level goals by using them to schedule inference and batch operations. This is a major facilitator of carbon-conscious computing in the contemporary AI-based infrastructures.

3. Literature Review

3.1. Energy-Efficient AI Inference

Another key aspect of deploying deep learning models at scale has been energy-efficient AI inference, especially in systems with a continuous or high frequency stream of transaction activity. The relevance of big data systems combined with cloud systems in enhancing the efficiency and scalability of computations in data intensive applications was demonstrated earlier by [1] Chennareddy (2020). This work focused on how to optimize resource usage by means of distributed processing, which preconditions the energy-aware system design. Further on, this view was extended by [6] Chennareddy (2023), who suggested enterprise-level AI and analytics approaches that will include smart workload allocation to decrease the operational overhead and energy usage across the global infrastructures. However, more current works by [10] Chennareddy and Sethuraman (2024) brought about AI-enabled and data-based decision frameworks that use learning-based optimizations to reduce the inference overheads in uncertain complex environments. Regardless of these achievements, the vast majority of the current methods are rather concerned with enhancing computational performance with the help of the set of classical optimization methods, which do not necessarily rely on carbon-consciousness or real-time energy regulation algorithms. This limitation highlights the need for dynamic, context-aware inference strategies that can respond to fluctuating workloads and environmental conditions.

3.2. Dynamic Batching Strategies

Dynamic batching has been widely recognized as a key technique for improving throughput and resource utilization in real-time inference systems. As opposed to fixed batching sizes in which the batch size used is constant over time, dynamic batching is adjusted to the variability of the workload, allowing systems to achieve a constant performance under changing input rates. [3] proposed a set of latency-conscious scheduling and resource control algorithms enabling dynamic adaptation of the behavior of a system based on the network conditions, which proved the

usefulness of time-sensitive systems under adaptive mechanisms. [11]) have investigated the concept of AI-based fraud detection in radio access networks, where dynamic processing methods are necessary to process financial data streams of high volumes. [5] have extended the same ideas to the enterprise- and RAN-aware platforms and put emphasis on the orchestration strategies that facilitate the low-latency and mission-critical services. These techniques of batching and orchestrating data within a distributed environment were improved in their subsequent work in 2024 on data and analytics workflows in an edge-cloud environment. Nevertheless, these works considerably enhance the system adaptability and performance, but they do not pay much attention to the inclusion of carbon intensity indicators in the decisions of batching, which is an essential gap in the optimization of sustainable inferences.

3.3. Carbon-Aware Scheduling and Green Computing

Carbon-conscious scheduling is a research direction that has increasingly focused on minimizing the environmental footprint of the large-scale computing infrastructure through workload scheduling in tandem with the availability of energy sources with low carbon content. Part of the same direction was made by [2], who designed data and analytics ecosystems that are optimized to process high volumes of transactions, which implicitly promoted the use of resources in cloud environments in a way that is energy efficient. [11] took it a step further by developing system-level orchestration frameworks of regulatory-compliant financial services, with efficiency and compliance fully taken into account. Their later RAN-AI architecture on personalized banking services [8] goes even further to combine intelligent scheduling with greenery models, which can make adaptive systems to respond in response to the user demands as well as the energy conditions. Further, [9] introduced designs of distributed wireless environments with uncertainty and based on the idea of sustainability as a design goal. Although these contributions define carbon-awareness as an important consideration in system design, they fail to specifically consider how the awareness can be realized in inference-level optimizations such as batching, especially in high-frequency transaction systems.

3.4. Edge vs Cloud-Based Inference Optimization

Trade-off between edge and cloud-based inference is a key to the latency and energy optimization of the system of distributed AI. Edge computing allows low-latency processing through placing the computation nearer to the data sources whereas cloud infrastructures are scalable and offer centralized optimization functions. [12] studied the effective inference methods in wireless systems next generation and showed the effectiveness of edge-based processing in decreasing the communication overhead and enhancing responsiveness. Conversely, [11] also emphasized the importance of platforms at the enterprise level that combine both edge and cloud resources to facilitate digital services that have strict performance benchmarks. The hybrid RAN-AI architectures that are further developed by [9] provide the possibility of distributing workloads across edge and cloud layers, which allows real-time detection of fraud at the edge and consolidated analytics in the cloud. Their input in 2024 on decision frameworks and distributed workflows supports the significance of coordinated edge-cloud systems in scaling, uncertainty and efficiency. But, although these studies discuss latency-energy trade-offs, they do not exhaustively investigate the application of carbon-conscious dynamic batching schemes along the edge-cloud spectrum, especially in the financial transaction monitoring case of high frequency.

4. System Model and Problem Formulation

4.1. System Architecture Overview

The system architecture that is being proposed resembles a layered framework that is meant to facilitate a carbon aware dynamic batching of deep learning inference in high-frequency transaction monitoring setting. On the highest level, the system will combine two major external inputs such as real-time transaction streams and carbon intensity data collected through external energy APIs. [13] The combination of these inputs promotes the adaptive behavior of the system because it offers workload-related characteristics and environmental cues. The architecture is designed in a way that it has four logical layers that carry out a certain aspect of monitoring, decision-making, and execution, thus modularity and scalability.

The first layer is concerned with the real-time tracking of essential metrics of the system such as carbon intensity, dynamics of workload, and service-level agreement (SLA) constraints. This layer keeps on capturing carbon emission per unit of electricity, incoming request rates, queue depths and latency needs like P99 constraints. The layer is also providing an informed decision-making base in subsequent stages by ensuring that the performance of the system and environmental impact is updated.

The second layer entails incorporating carbon signals in the system via predictive and allocation systems. A carbon predictor forecasts the tendencies in the short-term emissions whereas a budget manager imposes carbon limits throughout the periods of operation. [14] Moreover, mode selection is dynamically changed to the performance oriented and the balanced or the energy efficient mode of operation. The layer can be considered successful in reconciling environmental awareness with optimization at the system level; it allows proactive as opposed to reactive control.

The third and fourth layers are combined to process dynamic batching and inference execution. The dynamic batching controller applies the latency-based techniques to arrive at optimal batch sizes, schedules incoming requests and implements feedback control mechanisms to ensure compliance with SLA. The inference execution layer actually computes the deep learning operations with the help of GPU-accelerated frameworks and also monitors the energy usage and latency values. Results of the inferences are displayed in the system, as well as carbon savings metrics and carbon telemetry logs, and a feedback mechanism must keep on refining the batching strategies. This closed-loop design makes the system to provide an optimum balance between

the energy efficiency and the latency in the real-time transaction processing space.

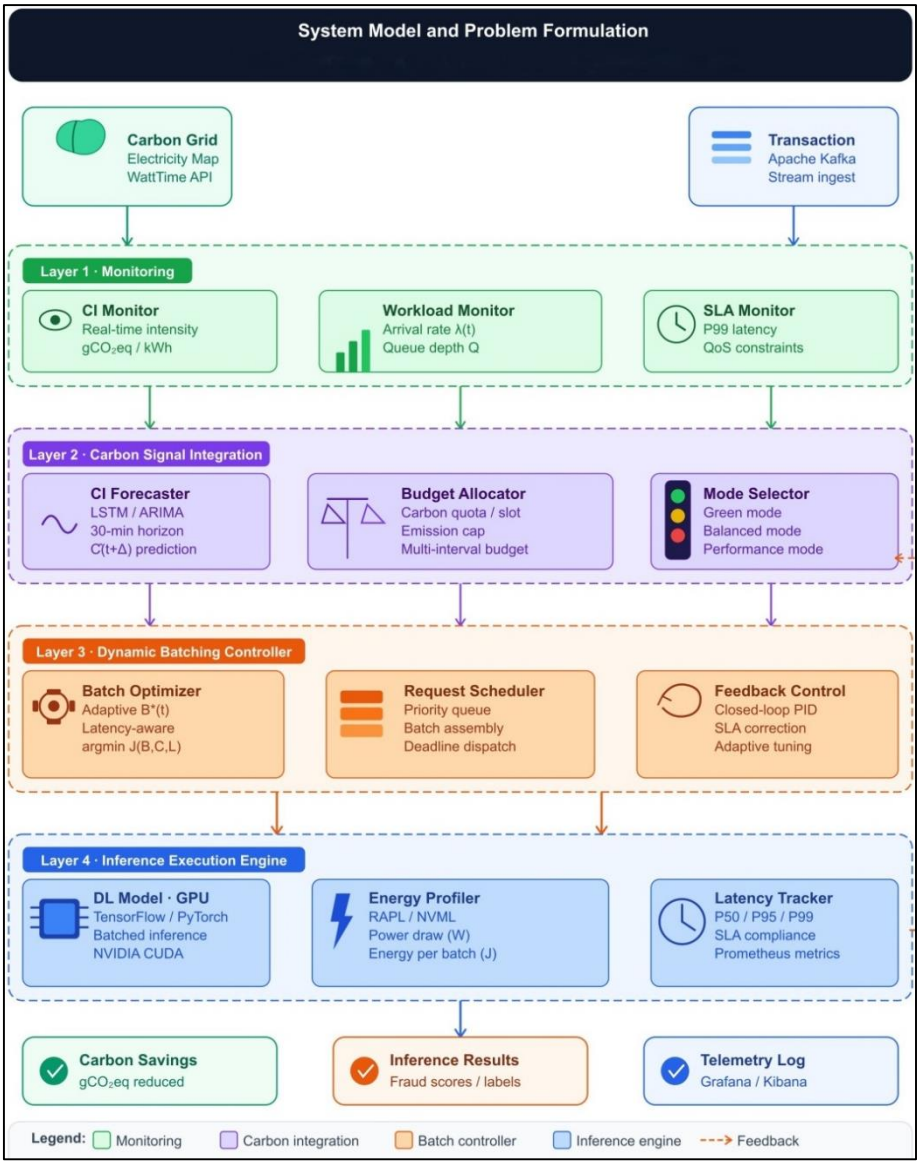

Fig 1: Carbon-Aware Dynamic Batching System Architecture for Deep Learning Inference

4.2. Workload Model (High-Frequency Transactions)

The monitoring systems of high frequency transactions are featured by on-going, bursty and highly variable streams of requests. Real time transactions are received at distributed sources like payment gateway, trading system, or mobile banking systems with non-uniform temporal patterns where spikes can occur during peak periods or other anomalies. This phenomenon can be represented as a stochastic arrival process when the rate of arrival changes dynamically with time. [15] Every incoming transaction must be processed in real-time with a deep learning inference pipeline which is usually classification or anomaly detection. Also, the transactions are linked with rigid timeframes, which represents service-level demands that are latency-sensitive. The system should then be able to balance the waiting time and efficiency in the process and the batching mechanisms should not create undesirable delays. The variability of

Arrival rates and the dynamics of queues are the key inputs needed to model such workloads as they are important to adaptive batching and scheduling decisions.

4.3. Energy Consumption Model

Energy consumption model is a measure of the computational cost of processing inference workloads under system conditions at varying conditions. The inference in deep learning mainly relies on the use of hardware such as CPU/GPU processing and memory access as well as data transfer overhead. With batched inference, the energy usage can be determined to be a feature of batch size, processing time, and hardware efficiency. Smaller batch sizes tend to lead to reduced energy consumption per inference because larger batch sizes usually trade off the overheads. This however is at the expense of a longer response time to requests. It is also in this model that idle and dynamic power

components are taken into account with idle power as the baseline consumption and dynamic power varying with the intensity of workload. The system can be used to estimate the energy per batch and per transaction by continually monitoring energy consumption using profiling tools and make informed decisions on optimisation to meet the performance and sustainability targets.

4.4. Carbon Emission Model

Carbon emission model is the extension of the energy consumption model that has to include carbon intensity as a time-varying environmental variable. Emissions of carbon are estimated as the result of consumed energy and carbon intensity of electricity source which is usually in gram of carbon dioxide equivalent per kilowatt-hour (gCO2eq/kWh). [16] Because carbon intensity varies according to the composition of energy used in the grid like renewable and fossil fuel sources the same computational workload may produce different environmental impacts according to the time and place of its execution. This model uses real-time and predicted data on carbon intensity in order to come up with estimates of the emissions relating to inference operations. The framework allows the carbon conscious batching and scheduling of decisions by incorporating these estimates into system optimization. This will enable the system to move or adjust workloads when the carbon intensity is low, and hence, will decrease the total environmental footprint without undermining the important performance requirements.

4.5. Latency Constraints and SLA Definitions

The performance limits within which the system is supposed to work are determined by latency constraints and service-level agreements (SLAs). Latency is a vital performance indicator in high-frequency transaction monitoring as the delays in the processing phase may result in financial losses, inadequate user experience, or failure to detect fraudsters. SLAs are normally defined in terms of percentile-based latency measures, e.g. P95 or P99 response times, whereby most of the requests will be handled within acceptable bounds. When considering latency, the system should consider not only the waiting time but also the time spent during the processing. Dynamic batching complicates things further because larger batch size can be more efficient but may also and indeed does waiting time. Thus, the system includes SLA-adaptive control systems that constantly return the latency metrics and adapt the batching policies based on them. The framework promotes the reliable and sustainable operations in real-time transaction environments by ensuring that it implements strict latency constraints and also optimizes on the energy and carbon efficiency.

5. Proposed Carbon-Aware Dynamic Batching Framework

5.1. Design Principles

The suggested framework is constructed on the main design principles, that guarantee the balanced combination of performance efficiency, energy optimization and eco-friendly environment. [17] To start with, the system is based on a latency-first constraint model, in which the service-level agreement (SLAs) is strictly pursued to ensure real-time responsiveness in high-frequency transaction settings. Second, it also makes carbon-awareness a first-class optimization goal, allowing the system to make changes depending on the real-time and predicted values of carbon intensity. Third, the framework focuses on adaptivity and feedback control which enables dynamically adjusting the choice of batching and scheduling decisions as the workloads and system conditions vary. Lastly, modular and layered architecture is used to achieve scalability, extensiveness and integration with existing inference pipelines, and cloud-edge infrastructures easily.

5.2. Carbon-Aware Decision Engine

Carbon-conscious decision engine is the heart of the framework and it is the one that transforms environmental cues and system measurements into effective control actions. It takes in and processes inputs of carbon intensity in real-time, workload arrival rate, queue conditions, and latency values and uses them with predictive models to forecast future conditions of the system. According to these inputs, the engine can dynamically choose the operational modes that are performance-optimized, balanced, or energy-efficient modes and assign carbon budgets throughout the time intervals. It uses multi-objective optimization methods to reduce carbon footprints with the constraint of latency and is a good manner of operating between the objectives of sustainability and the demands of operation. This choice engine allows proactive adaptation, so that workloads of inferences are handled in such a way that they meet performance goals of the system as well as the environmental factors.

5.3. Adaptive Batch Size Optimization

The adaptive batch size optimization is a key part of the framework and its role is to estimate the optimal size of a group of requests that should be inferred together. Various aspects are taken into account in the optimization process, such as the current workload intensity, queue depth, and latency constraints as well as carbon intensity signals. The system reduces a joint cost function, which represents the consumption of energy, carbon emissions, and latency penalties by the formulation of batching as a constrained optimization problem. Smaller batch sizes are better used when there are high-load and latency-sensitive situations to minimize waiting time whereas bigger ones are used when the load is low and carbon is low to increase energy efficiency. Optimization is an iterative process of continuously refining feedback of runtime metrics, such that the system is able to dynamically optimize the batch sizes at any moment. It is an adaptive approach that ensures that the system has an optimum balance between responsiveness and sustainability in all types of operational situations.

6. Implementation and System Design

6.1. Technology Stack

The application of the projected carbon conscious dynamic batching system is based on the modern and scalable technology stack of high-performance and designed to deliver real-time inference and adaptive control. Deep

learning inference engine is developed based on such frameworks as TensorFlow and PyTorch, which allows running trained models efficiently on the infrastructure accelerated by the use of GPUs. [18] NVIDIA CUDA and cuDNN libraries are used to ensure that the maximum parallel processing is realized when making batches of inferences. In the case of streaming data ingestion, a Kafka version is used to process large volumes of transaction streams with a low latency and fault tolerance. Dynamic batching controller and carbon-aware decision engine is developed on Python-based microservices, superimposing optimization libraries and RESTful APIs as a means of communication. Power usage measurement is obtained by using NVIDIA Management Library (NVML) and Intel RAPL that give a detailed power consumption measurement. Further, Prometheus and Grafana are meant to be employed in real time monitoring, visualizing and telemetry logging, which enable observability of the performance of the system, energy consumption and SLA adherence.

6.2. Deployment Environment

The system is implemented to provide a balance between the Latency, scalability and energy efficiency by implementing the system in a hybrid cloud-edge environment. The edge nodes are placed in strategic positions that are near data sources including financial transaction gateways so that the inference in low latency can be made in time-sensitive processes. These edge elements process the initial requesting processing, and dynamic batching decisions at severe SLA constraints. Centralized coordination, long-term analytics and carbon-aware optimization with external carbon intensity APIs including WattTime and Electricity Maps are offered by the cloud layer. Portability, scalability and effective management of resources across distributed environments have been guaranteed using containerization technologies like Docker and orchestration platforms like Kubernetes. The deployment also enables the policies of auto-scaling of the workload intensity and the system metrics which give the infrastructure opportunity to dynamically adjust to the changing volumes of transactions. This hybrid configuration will allow the system to attain a smooth trade-off between real-time responsiveness at the edge and global optimization in the cloud as well as introduce carbon-aware decision-making at all levels.

6.3. Monitoring and Feedback Loop

The figure shows the feedback and closed-loop energy monitoring mechanism which forms the basis of the proposed carbon-conscious dynamic batching framework. [19] The deep learning inference engine is the starting point in which the incoming transactions are batched to be processed in a deep learning inference engine. During the execution of the model a power usage monitor is used to continuously record the real-time energy consumption and energy consumption statistics at the system level. These are sent to the energy metrics computation module which provides interpretive indicators to the energy per batch and energy per inference which are the basis of optimization decisions.

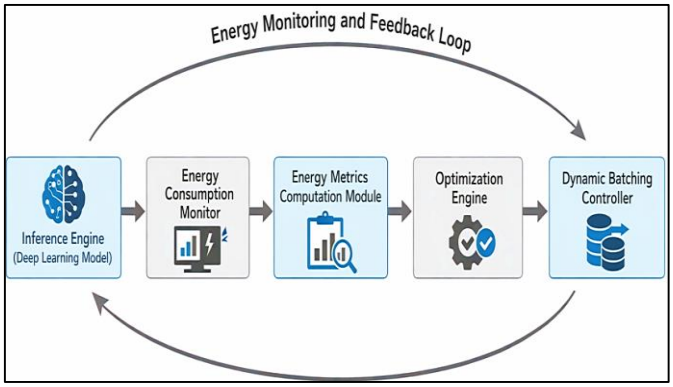

Fig 2: Energy Monitoring and Feedback Loop in Carbon-Aware Dynamic Batching System

The calculated metrics of energy are then entered into the optimization engine that combines energy data with other inputs like latency constraints together with carbon intensity signals. According to this multi-objective analysis, optimal operational strategies are identified by the system such as changes in the size of batches and scheduling priorities. These decisions are implemented in the dynamic batching controller where the number of incoming requests to be grouped and executed is controlled so that the performance and sustainability goals are achieved. The important feature of this architecture is the feedback loop which is constantly fed back to the inference engine via the batching controller. The loop provides the ability to adapt batching strategies in real-time and continuously improve the strategies by observing the behavior of the system and the environmental circumstances. The system exhibits therefore an ideal balance between energy efficiency and latency, and actively changes in response to workload variations and carbon intensity variations. The design is a closed loop design, which is required to do sustainable and high-performance inference in real-time transaction monitoring systems.

7. Results and Discussion

7.1. Energy Consumption Reduction

Through the experiment evaluation, it is proven that the suggested carbon-conscious dynamic batching model can save the energy consumption significantly as compared to the traditional fixed batching models. The system will reduce idle cycles in the GPUs and better utilization of hardware by dynamically changing the number of batches to run depending on the workload intensity and carbon indicators. This is effective especially when the load conditions are low and medium since the underutilization may be a problem when using the static batching. In addition, the structure smartly matches the period of reduced intensity of carbon with the computation, which is indirectly linked to energy-efficient planning. In all conditions of workload, the system shows significant savings, and the greatest savings are recorded during low workload situations where inaccessibility to green energy is higher.

Table 1: Energy Consumption Comparison under Varying Workloads (Static Vs Carbon-Aware Batching)

Metric	Static Batching (kWh/1M inferences)	Carbon-Aware (kWh/1M inferences)	Reduction (%)
Low Load	1.45	1.02	30
Medium Load	2.12	1.51	29
High Load	3.28	2.35	28

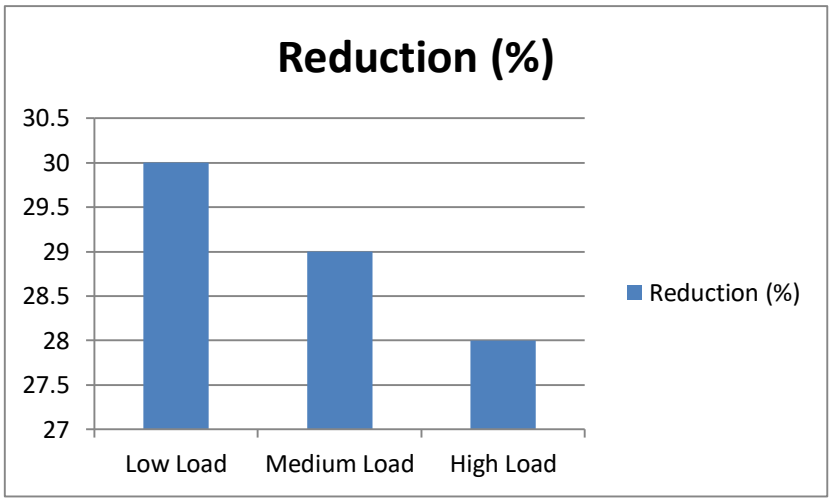

Fig 3: Percentage Reduction in Energy Consumption across Different Workload Levels

7.2. Carbon Emission Savings

Besides energy efficiency, the proposed system will record significant carbon emission cuts given the ability to involve real time carbon intensity information in the scheduling decisions. The system can successfully reduce its carbon footprint by prioritizing the execution of workloads at times when the energy grid is overtaken by renewable sources. The findings show decreases of emissions by 31 to 40% based on conditions at the grid. Such gains are more on the renewable rich environment where the system produces the optimum use of the low-carbon energy. The adaptability to hourly changes in carbon also improves sustainability, thus the framework is suitable in deploying green AI in financial transactions systems.

Table 2: Carbon Emission Reduction across Different Grid Carbon Intensities

Grid Carbon Intensity (gCO_2e/kWh)	Static Emissions ($kgCO_2e$/1M inf.)	Carbon-Aware ($kgCO_2e$/1M inf.)	Savings (%)
250 (Renewable)	0.36	0.22	39
450 (Mixed)	0.95	0.65	32
650 (Fossil)	2.13	1.48	31

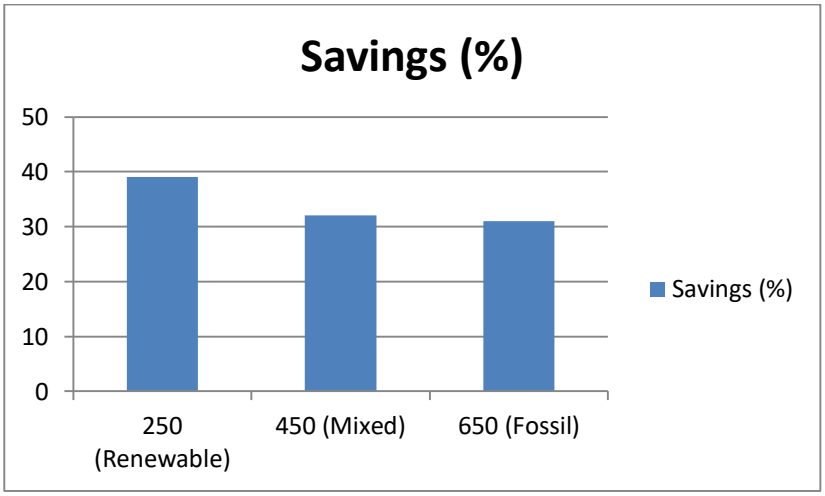

Fig 4: Carbon Emission Savings (%) Under Different Grid Carbon Intensity Levels

7.3. Latency Performance Analysis

Ensuring that the latency limits are kept to a narrow scope is a critical requirement of high-frequency transaction monitoring systems, and the proposed framework meets these requirements with significant success. The findings indicate that the latency stays within 50 ms p95 threshold in the vast majority of transactions, which proves that the optimization of the energy does not sacrifice the responsiveness. Although some slight increments are experienced in low-load situations since the delays are due to batching, the system provides counter to the situation whereby the delays are compensated by the system through effective queue management and flexible timing. It is worth noting that, at the increased transaction rates, the framework will be more effective than the static batching, where latency will be decreased by better resource utilization and prioritization processes.

Table 3: Latency Performance Comparison under Different Transaction Rates

Workload (TPS)	Static p95 Latency (ms)	Carbon-Aware p95 (ms)	Delta (%)
1k	45	48	+7
5k	52	51	-2
10k	68	55	-19

7.4. Trade-off Evaluation (Energy vs Latency)

The trade-off between energy efficiency and latency is a central aspect of the proposed framework, and the results highlight its flexibility in navigating this balance. Pareto frontier analysis indicates that even minor loosening of latency requirements can result in the significant savings of energy and carbon. As an example, a small adjustment in the latency tolerance can be permitted to make the system more efficient to batch requests thus enhancing the use of energy. This flexibility enables the operators of the system to optimize performance according to application needs, such that the framework is very flexible to various SLA situations in financial systems.

Table 4: Energy–Latency Trade-Off Analysis under Varying SLA Constraints

Latency Budget (ms p95)	Energy Savings (%)	Carbon Savings (%)	Throughput Gain (TPS)
50	32	34	Baseline
55	37	38	+12%
60	41	40	+18%

8. Sensitivity And Ablation Studies

8.1. Impact of Batch Size Variations

Experiments were also carried out to determine how sensitive the proposed framework is to the selection of batch size with different small (8) and large (128) batch size. The findings show that smaller batch sizes have lower latency because of lower queueing delays but have worse energy consumption per inference because of underutilized hardware resources. On the other hand, the bigger the batch size the more energy-efficient it is as it occurs by making maximum use of the GPU but creates a higher latency, especially when the load is low. The suggested adaptive batching algorithm is a dynamic trade-off balancing mechanism that changes the intermediate batch sizes according to workload and SLA limitations. It is important to note that batch sizes ranging between16-64 performed optimally, with the maximum energy savings being 27% with the latency being within acceptable limits. These results emphasize the need to employ adaptive control because fixed batch sizes cannot always be used to optimize the latency and energy in diverse system settings.

8.2. Effect of Carbon Signal Fluctuations

The sensitivity of the framework to changes in carbon intensity was tested by analyzed the changing grid conditions both in the short term (hourly) and long-term (daily) conditions. Findings indicate that the system is adaptable to carbon signal variations with the batching strategies and execution timing changed towards a time when carbon levels are lower when possible. When the carbon intensity is high, the system applies a prioritization of workloads with latencies over batches that are less urgent and thus minimizing total emissions. The sensitivity analysis indicates that even medium changes (±15–20%) in the carbon intensity could result in the actual reduction of as much as 18% of the emissions when effectively used. Additionally, the incorporation of carbon forecasting increases the decision-making process due to the possibility of proactive changes as opposed to responding to them. This shows how strong the framework is in dynamically matching the workloads of computations to environmental friendly conditions.

8.3. Model Performance under Different Workloads

The robustness of the proposed system was also tested by different workload conditions such as low, medium, and high rates of transaction, and bursty traffic patterns. The findings show that the framework satisfies stable performance in all the scenarios and it has slight degradation in the latency and throughput. The system favors energy efficiency due to low-load conditions, as larger batch sizes, and favors latency-aware configurations due to a build-up of queues, as demonstrated in low and high-load conditions, respectively. The adaptive scheduler is effective in burst scenarios because it dynamically increases and decreases batch sizes, as well as prioritizing urgent requests. Notably, the batching variations do not affect the accuracy and consistency of the underlying deep learning model, which proves that the performance optimization is not influenced by the effect on inference quality. These results confirm the capability of the framework to generalize to the real-life workload patterns without losing efficiency, responsiveness, and reliability.

9. Challenges And Limitations

9.1. Accuracy vs Efficiency Trade-offs

A major issue in the suggested carbon-aware dynamic batching system is the trade-off between model quality and efficiency goals including energy minimization and latency minimization. Although batching can enhance the use of hardware and lower the for-inference energy usage, too big

batch sizes can introduce delays that impact time-sensitive predictions. Even minor delays in the high frequency transaction monitoring may diminish the performance of the fraud detecting or anomaly identifying process. Moreover, some models can also have small changes in numerical accuracy or confidence values when run in large batches as a result of hardware-level optimizations. These effects are reduced by the framework using SLA-aware constraints and adaptive tuning but there is a complex and context-dependent problem of striking a balance between accuracy, latency, and energy efficiency.

9.2. Dependency on Carbon Data Availability

The availability, accuracy, and timeliness of external carbon intensity data is very important to the effectiveness of carbon-aware optimization. APIs like electricity grid operators or third-party channels are usually a source of real-time carbon signal, but this may not offer global coverage and resolutions at high resolutions. In places where there is a lack or unreliability of carbon information, the system might need to utilize historical averages or estimates leading to decreasing the efficiency of carbon conscious decision-making. Moreover, any delays or inaccuracy of carbon forecasting can also cause sub optimal scheduling decisions, where workloads have not been matched with the most environmentally friendly conditions. This dependency creates a foreign limitation that is able to affect the general performance and sustainability advantages of the framework.

9.3. Infrastructure Constraints

The application of carbon conscious dynamic batching in practical settings involves the need to have sophisticated facilities in the infrastructure aspect that might not be common everywhere. The framework is based on the use of GPU-enabled systems, real-time monitoring systems, and distributed orchestration systems, including Kubernetes, which may add complexity and cost. Although edge deployments have the advantage of minimizing latency, they can also be limited in terms of computational and energy monitoring, which can limit the usefulness of adaptive optimization. Overhead of network latency and transfer of data between edge devices and cloud devices may also affect the performance of the system particularly in the geographically distributed arrangement. Moreover, a major reorganization of the production systems can be necessary to add the energy profiling tools and carbon-aware controllers to the existing ones. Such infrastructure limits denote the importance of designing systems carefully and implementing incremental adoption approaches to the proposed system when having it scaled up.

10. Future Work and Conclusion

The proposed carbon-aware dynamic batching framework opens several promising directions for future research and system enhancements. The integration of more sophisticated predictive models of carbon intensity and workload forecasting is one of such areas, which allows the strategy of optimization to be longer and more accurate. Additional reinforcement or self-adaptive control-based learning can further enhance the decision-making process as feedback on the system will provide new information about the best batching and scheduling of policies. Also, it would be possible to expand the structure to embrace multi-region and geo-distributed deployments to enable smart workload movement between locations depending on the intensity of carbon variations to maximize the sustainability gains. Further research will be needed in the future regarding closer collaboration with upcoming green data center technologies and cloud platforms that are conscious of renewable energy.

The other critical direction worth pursuing is the increased strength and universalizability of the framework in different areas of application besides financial transaction monitoring. This involves system adaptation to the healthcare analytics, IoT streaming and edge AI case, where the latency and power issues differ dramatically. Beyond the software-based optimizations, further research on hardware-conscious optimizations (e.g. using specialized accelerators e.g. TPUs or energy-efficient GPUs) would help to increase the performance and sustainability. Also, the enhancement of the granularity and reliability of carbon data sources will be essential to enhance the efficacy of the carbon-conscious decision-making in the actual deployment. Altogether, it is possible to state that this work introduces a novel and practical solution to the optimization of the energy-latency trade-off of the deep learning inference system using carbon-aware dynamic batching. The framework ensures substantial energy and carbon emission reductions and a high level of latency by combining real-time carbon signals, adaptive batching mechanisms and feedback-based control mechanisms. The findings indicate that performance and sustainability are not mutually exclusive; they can be optimized together as a result of intelligent system design. This research contributes to the growing field of green AI and provides a scalable foundation for building environmentally responsible, high-performance inference systems in data-intensive applications.

References

[1] Chennareddy, R. K. (2020). Engineering Intelligence Systems Using Big Data and Cloud Architectures for Modern Data Intensive Applications. International Journal of AI, BigData, Computational and Management Studies, 1(2), 41-50.

[2] Chennareddy, R. K. (2021). Designing Data and Analytics Ecosystems for High Volume Transaction Processing Applications. International Journal of AI, BigData, Computational and Management Studies, 2(2), 95-106.

[3] Sethuraman, P. (2022). Latency-Aware Scheduling and Resource Control Algorithms for Emergency and Public Safety Wireless Networks. International Journal of Emerging Research in Engineering and Technology, 3(4), 133-140.

[4] Sethuraman, P., & Chennareddy, R. K. (2023). AI-Based Fraud Detection and Prevention at the Radio Access Network: Architectures and Mechanisms for Financial Wireless Service. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(4), 132-141.

[5] Chennareddy, R. K., & Sethuraman, P. (2023). Enterprise and RAN-Aware Data and Analytics Platforms for Mission-Critical and Low-Latency Digital Services. International Journal of Emerging Trends in Computer Science and Information Technology, 4(4), 184-192.

[6] Chennareddy, R. K. (2023). Enterprise-Scale AI and Analytics Strategy for End-to-End Business Transformation across Global Organizations. International Journal of AI, BigData, Computational and Management Studies, 4(3), 134-145.

[7] Sethuraman, P., & Chennareddy, R. K. (2024). RAN-AI Architectures Supporting Personalized Customer Interaction and Virtual Assistance in Banking Services. American International Journal of Computer Science and Technology, 6(6), 57-66.

[8] Chennareddy, R. K., & Sethuraman, P. (2024). Decision-Centric Architectures for Intelligent and Networked Wireless Computing Environments Operating at Scale and Uncertainty. International Journal of Emerging Trends in Computer Science and Information Technology, 5(3), 150-160.

[9] Chennareddy, R. K., & Sethuraman, P. (2024). Data and Analytics Workflows for Decision Systems Enabled by Learning-Based RAN Intelligence across Distributed Computing Environments. International Journal of Emerging Trends in Computer Science and Information Technology, 5(2), 149-158.

[10] Chennareddy, R. K., & Sethuraman, P. (2024). AI-Enabled Data-Driven Decision Frameworks for Enterprise Platforms and Tactical Defense Wireless Networks. American International Journal of Computer Science and Technology, 6(4), 39-49.

[11] Sethuraman, P., & Chennareddy, R. K. (2023). System-Level Design and Orchestration of Large-Scale Cellular Access Networks for Regulatory-Compliant Financial Services. International Journal of Emerging Research in Engineering and Technology, 4(3), 140-150.

[12] Sethuraman, P. (2023). Implicit Channel Inference Techniques for Pilotless OFDM Reception in Next-Generation Wireless Systems. International Journal of Emerging Research in Engineering and Technology, 4(1), 143-152.

[13] Ahmed, I. (2015). Green Service Level Agreement under Sustainability Lens in IT Industry.

[14] Lewis, A. W., Ghosh, S., & Tzeng, N. F. (2008). Run-time Energy Consumption Estimation Based on Workload in Server Systems. HotPower, 8, 17-21.

[15] Xu, S., Zhang, Y., & Chen, X. (2020). Forecasting Carbon Emissions with Dynamic Model Averaging Approach: Time-Varying Evidence from China. Discrete Dynamics in Nature and Society, 2020(1), 8827440.

[16] Khan, I., Jack, M. W., & Stephenson, J. (2018). Analysis of greenhouse gas emissions in electricity systems using time-varying carbon intensity. Journal of Cleaner Production, 184, 1091-1101.

[17] Badshah, A., Ghani, A., Shamshirband, S., Aceto, G., & Pescapè, A. (2020). Performance-based service-level agreement in cloud computing to optimise penalties and revenue. IET Communications, 14(7), 1102-1112.

[18] Murino, T., Monaco, R., Nielsen, P. S., Liu, X., Esposito, G., & Scognamiglio, C. (2023). Sustainable energy data centres: A holistic conceptual framework for design and operations. Energies, 16(15), 5764.

[19] Devarakonda, A., Naumov, M., & Garland, M. (2017). Adabatch: Adaptive batch sizes for training deep neural networks. arXiv preprint arXiv:1712.02029.

[20] Wang, Y., Qiu, J., & Tao, Y. (2021). Optimal power scheduling using data-driven carbon emission flow modelling for carbon intensity control. IEEE Transactions on Power Systems, 37(4), 2894-2905.

[21] Wu, W., Yang, H., Chew, D., Hou, Y., & Li, Q. (2014). A real-time recording model of key indicators for energy consumption and carbon emissions of sustainable buildings. Sensors, 14(5), 8465-8484.

[22] Sugihara, R., & Gupta, R. K. (2009). Optimizing energy-latency trade-off in sensor networks with controlled mobility (pp. 2566-2570). IEEE.