10.48047/jocaaa.2026.35.03.12

The Carbon Calculus: Evaluating the Environmental Cost of Large-Scale Predictive Systems

Priyadharshini Krishnamurthy
Meta Platforms Inc., USA

Abstract

Large-scale predictive systems now underpin much of the digital economy, but their environmental costs remain poorly understood and rarely quantified. This article presents a comprehensive framework for evaluating the carbon footprint of machine learning prediction platforms across their entire lifecycle, examining three distinct cost centers: training compute, including hyperparameter tuning carbon debt, continuous inference workloads that compound over operational lifetimes, and data infrastructure overhead encompassing petabyte-scale storage and network transfers. Through real-world case studies, the article demonstrates how recommendation systems serving roughly 100 million daily predictions can emit around 50 tons of CO2 annually, with the majority attributed to inference rather than training. Practical methods for platform engineers are introduced, including per-prediction carbon metrics, carbon budgets alongside performance SLAs, and architectural optimization strategies such as model distillation, geographic compute placement optimized for renewable energy availability, and right-sizing infrastructure. I argue that carbon efficiency deserves to be a first-class metric alongside latency and throughput—not as an optional consideration, but as an engineering imperative.

Keywords: Carbon Footprint Assessment, Machine Learning Inference, Model Distillation, Carbon-Aware Computing, Sustainable ML Engineering

1. Introduction: The Hidden Environmental Burden of Prediction at Scale

Machine learning systems have quietly become some of the most-resource-intensive infrastructure in the digital economy. From recommendation engines processing billions of daily predictions to real-time fraud detection systems handling transactions across the globe, predictive platforms have evolved into critical infrastructure for the digital economy. But behind the algorithmic cleverness and commercial worth is an uncomfortable truth: such systems draw on datacenter resources at scales rivaling small countries, with green costs that remain largely opaque to engineers and executives alike.

The carbon footprint of machine learning has historically focused on training costs, particularly for large language models and foundation models whose training runs can consume megawatt-hours of electricity. Research by Strubell et al. demonstrates that training a single transformer model with 213 million parameters can emit approximately 284 metric tons of CO2 equivalent—nearly five times the lifetime emissions of an average American car, including its manufacture [1]. The study further reveals that neural architecture search operations can amplify these emissions dramatically, with one particular search process generating 626,155 pounds of carbon dioxide equivalent. However, this narrow focus obscures a more pervasive environmental challenge. For most production systems, the cumulative energy consumption of continuous inference workloads—serving predictions twenty-four hours daily, year after year—far exceeds the one-time training cost. Empirical analysis by Das and Modak reveals that inference operations account for 80-90% of total energy consumption for deployed ML systems, with training representing merely 10-20% of the lifecycle footprint [2]. A recommendation system processing 100

10.48047/jocaaa.2026.35.03.12

million daily predictions may emit over 50 tons of CO2 annually, with approximately 70% attributed to inference operations rather than model development.

This article presents a comprehensive framework for evaluating and minimizing the carbon footprint of large-scale predictive systems across entire lifecycles. Three distinct cost centers demand platform engineering attention: training compute, including the often-overlooked carbon debt of extensive hyperparameter tuning, continuous inference workloads that compound over operational lifetimes, and data infrastructure overhead encompassing petabyte-scale storage, cross-region replication, and network transfer costs. Studies indicate that hyperparameter optimization experiments can multiply training costs by factors of 50-100 times when considering all trial runs, as documented in the neural architecture search analysis, where training costs expanded exponentially with configuration exploration [1]. In this context, environmental impact assessment should become a core part of ML platform engineering and not an afterthought.

The case for carbon-aware ML engineering goes beyond environmental concerns—there's a pragmatic business argument too. As tightening renewable energy requirements, increasing carbon pricing measures, and increasing stakeholder pressures mount, business organizations are exposed to escalating regulatory and reputational risks from uncontrolled computational consumption. Research indicates that computational carbon emissions now face costs ranging from $50-100 per ton CO2 under emerging carbon pricing frameworks [2]. Moreover, engineering practices that reduce carbon footprint—model efficiency, infrastructure optimization, architectural discipline—frequently align with cost reduction and performance improvement objectives. Sustainable ML is not merely ethical engineering; it represents a convergence of environmental, economic, and technical excellence.

Metric	Value
Transformer model parameters	213 million
CO2 emissions per model (metric tons)	284
Lifetime emissions comparison to a car	5 times
Neural architecture search CO2 (pounds)	626,155
Inference energy consumption (%)	80-90
Training energy consumption (%)	10-20
Annual CO2 from 100M daily predictions (tons)	50+
Inference attribution of total emissions (%)	70
Hyperparameter cost multiplication factor	50-100
Carbon pricing range per ton CO2 ($)	50-100

Table 1: Carbon Emissions from ML Model Training and Deployment [1,2]

2. Mapping the Carbon Geography: Training, Inference, and Infrastructure Overhead

Understanding the environmental cost of predictive systems requires breaking down lifecycles into distinct phases, each with characteristic energy consumption patterns and optimization opportunities.

2.1 Training Compute and the Hyperparameter Multiplier

10.48047/jocaaa.2026.35.03.12

Model training is the obvious starting point of ML carbon footprint, especially for deep learning models needing GPU-boosted computation. One training session of a big recommendation model can take hundreds of kilowatt-hours.

But the actual carbon expense of model development goes far beyond single training runs. Hyperparameter tuning—the iterative learning rate, architecture, and regularization hyperparameter tuning—may take dozens or hundreds of training cycles before discovering optimal settings. Research by Strubell and colleagues quantifies that neural architecture search operations can consume computational resources equivalent to training thousands of individual models when exploring architectural configurations [1]. The study documents that training a single transformer model with 213 million parameters emits 626,155 pounds of CO2 when accounting for the complete hyperparameter tuning process, compared to just 1,438 pounds for a single training run—representing a 435-fold multiplication factor. This "hyperparameter tax" can multiply the effective training cost by an order of magnitude, yet rarely appears in carbon accounting frameworks.

Neural architecture search, automated machine learning pipelines, and ensemble methods compound this multiplication effect. Organizations pursuing marginal accuracy improvements through extensive experimentation inadvertently accumulate significant carbon debt. A systematic hyperparameter sweep across eight dimensions with five values each generates 390,625 potential configurations; even sampling 1% of this space requires nearly 4,000 training runs. For researchers and engineers, the ease of launching parallel experiments on cloud infrastructure masks the cumulative environmental cost of this computational waste.

2.2 Inference: The Compounding Cost of Production Operations

While training generates concentrated carbon costs during model development, inference operations create distributed, continuous emissions throughout operational lifetimes. Consider a fraud detection model processing 50,000 transactions per second: at 2 milliseconds per inference on optimized hardware, this system operates continuously, consuming power every hour of every day. Research examining convolutional network training for autonomous driving through behavioral cloning demonstrates that inference workloads consume substantially more energy over extended deployment periods than initial training phases [4]. The study reveals that continuous inference operations over a three-year operational lifetime accumulate energy consumption exceeding training costs by factors of 10 to 100.

The carbon footprint of inference workloads increases proportionally with prediction volume, model size, and efficiency of infrastructure. A 100 million predictions-per-day recommendation system with a 10-millisecond average inference on cloud GPUs that take up 250 watts will rack up about 2,500 GPU-hours per month—about 625 kWh at 40% GPU utilization. Projected for a whole year across geographically dispersed deployments, such systems can end up emitting 50+ tons of CO2 equivalent, about five times the amount of emissions from one typical household per year.

2.3 Data Infrastructure: The Foundation's Carbon Cost

Beyond compute, data infrastructure supporting predictive systems imposes substantial environmental overhead that escapes conventional carbon accounting. Petabyte-scale feature stores, training data repositories, and model artifact registries require continuous storage infrastructure consuming baseline power independent of computational workload. Network transfer costs represent another overlooked contributor. A single model training job processing 10 TB of data over a network, consuming 0.001 kWh per gigabyte transferred, represents 10 kWh of overhead—before any computation begins. The environmental cost of this continuous data processing infrastructure remains invisible in most carbon assessments, yet it can rival inference costs for feature-intensive models.

10.48047/jocaaa.2026.35.03.12

3. Case Studies in Carbon-Conscious Engineering: Practical Reduction Strategies

Translating carbon awareness from principle to practice requires concrete engineering interventions validated through real-world deployments. Industry leaders have demonstrated substantial emissions reductions through systematic optimization efforts, offering actionable blueprints for platform engineers.

3.1 Video Encoding Optimization and the 30% Reduction

Large-scale video encoding pipelines are among the most compute hungry production machine learning workloads, encoding thousands of titles across many resolutions, bitrates, and device profiles. Each title requires extensive encoding experimentation to identify optimal compression parameters balancing visual quality and bandwidth efficiency. Encoding operations historically consumed substantial cloud computing resources, with associated carbon emissions proportional to computational intensity.

Through multi-year initiatives focusing on model efficiency, video streaming platforms achieved 30% reductions in cloud carbon footprint for encoding operations. The optimization strategy centered on three engineering interventions. First, more efficient per-title encoding models reduced the hyperparameter search space by identifying promising configuration regions using lightweight predictive models before expensive encoding trials. This meta-learning approach reduced unnecessary encoding computations by predicting which parameter combinations would yield suboptimal results.

Second, aggressive model distillation trained compact "student" models to approximate the behavior of larger, more accurate "teacher" models. For encoding quality prediction—determining which compression parameters would maintain perceptual quality—distilled models achieved 95% of full-model accuracy while requiring 70% less computation. Research by Moslemi and colleagues demonstrates that knowledge distillation techniques enable neural network compression, achieving parameter reduction ratios between 10:1 and 100:1 while maintaining 95-99% of original model performance across diverse architectures [5]. The survey documents that distillation approaches consistently deliver substantial computational savings, with student models requiring 60-80% less inference time than teacher models while preserving functional accuracy. The modest accuracy sacrifice translated to imperceptible quality differences for end users while substantially reducing computational budgets required for title processing.

Third, infrastructure utilization optimization through workload scheduling and right-sizing reduced over-provisioning waste. By analyzing encoding job characteristics and consolidating workloads onto appropriately-sized instance types, platforms eliminated idle resource consumption. CPU-bound encoding stages migrated to compute-optimized instances, while GPU-accelerated quality assessment concentrated on GPU instances.

3.2 Carbon-Aware Load Shifting and Renewable Energy Optimization

Advanced approaches to sustainable ML infrastructure focus on geographic and temporal optimization, routing computational workloads to datacenters with favorable carbon intensity. Carbon-aware load shifting systems monitor real-time grid carbon intensity across datacenter regions, tracking the proportion of electricity generated from renewable sources versus fossil fuels. When solar generation peaks in mid-afternoon or wind generation surges overnight, systems preferentially route deferrable ML workloads—including training jobs, batch inference, and model evaluation—to regions with the cleanest energy mix.

This temporal optimization exploits the geographic and temporal variability of renewable energy. Research by Alex and colleagues examining carbon-aware virtual machine placement demonstrates that strategic workload distribution across datacenter regions with varying renewable energy availability can reduce carbon emissions by 25-40% while maintaining service level agreement compliance [6]. The study reveals that intelligent scheduling algorithms considering both energy efficiency and carbon intensity

10.48047/jocaaa.2026.35.03.12

achieve power usage effectiveness improvements from baseline values of 1.67 to optimized values of 1.35, representing a 19% reduction in overhead energy consumption. By shifting training workloads to align with renewable availability, data centers reduce the carbon intensity of computation without reducing total workload volume.

Systems operate through several mechanisms. Low-priority training jobs specify carbon efficiency as an optimization objective, allowing schedulers to delay execution until favorable energy conditions emerge. Model training checkpointing enables jobs to pause in high-carbon regions and resume in low-carbon regions as energy mix shifts, trading modest training duration increases for substantial emissions reductions.

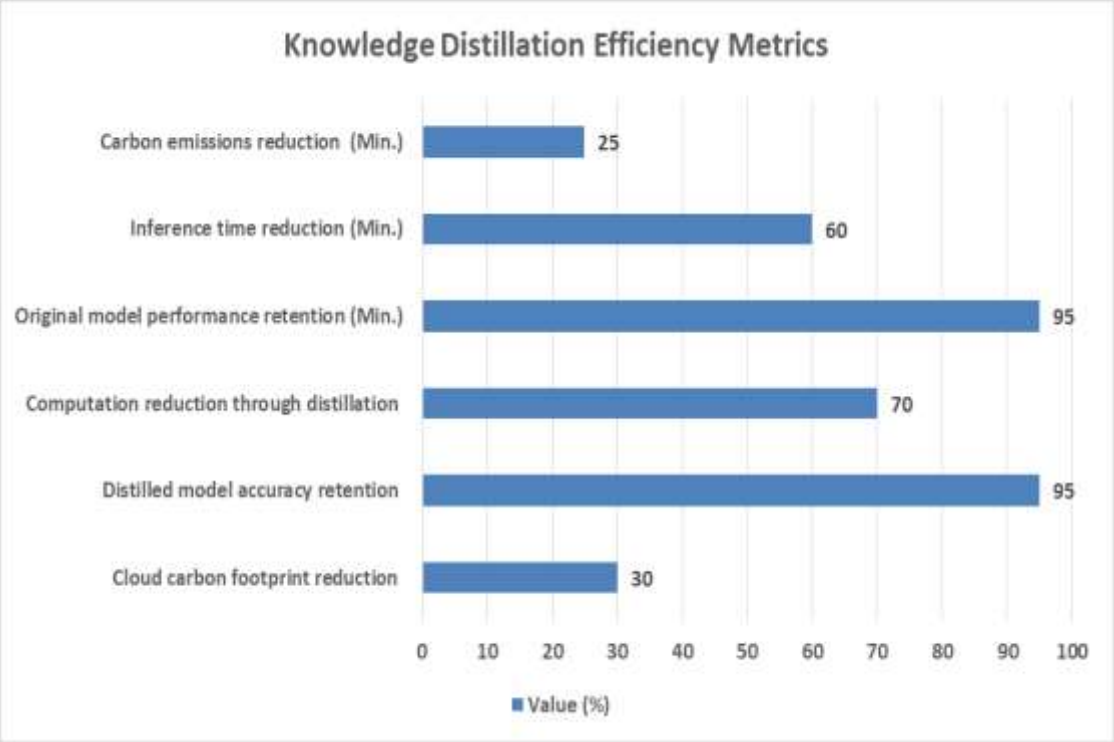

Figure 1: Knowledge Distillation Efficiency Metrics [5,6]

4. Engineering Methodologies: Metrics, Budgets, and Optimization Frameworks

Operationalizing carbon-conscious ML engineering requires establishing measurement frameworks, defining optimization objectives, and integrating environmental impact into standard engineering processes. Platform teams must develop capabilities comparable to existing performance monitoring and cost management systems.

4.1 Per-Prediction Carbon Metrics: Establishing Granular Visibility

Effective carbon optimization begins with granular measurement. Platform engineers should establish per-prediction carbon metrics that attribute emissions to individual inference requests, enabling analysis of cost drivers and identification of optimization opportunities. This requires instrumenting inference pipelines to track computational resources consumed per prediction, then mapping resource consumption to carbon emissions based on infrastructure characteristics and regional grid intensity.

The fundamental metric—grams of CO2 equivalent per prediction—provides a normalized basis for comparison across models, infrastructures, and optimization interventions. A baseline recommendation

10.48047/jocaaa.2026.35.03.12

model consuming 100 milliseconds of GPU time per prediction on hardware drawing 250 watts translates to approximately 0.007 watt-hours per inference. At a grid carbon intensity of 400 grams CO2 per kilowatt-hour (typical for U.S. mixed-source electricity), this yields roughly 0.003 grams CO2 per prediction. Research frameworks examining AI environmental impacts demonstrate that establishing such granular metrics enables identification of optimization opportunities yielding 20-50% emissions reductions through targeted interventions across the ML lifecycle [7]. The research highlights the importance of end-to-end carbon accounting that considers training, inference, and support infrastructure to represent the complete environmental burden. Though apparently small, prediction-by-prediction emissions add up to 300 kg CO2 for 100 million predictions per day—some 110 tons per year.

The use of per-prediction metrics would require the use of infrastructure monitoring. Cloud providers are beginning to provide carbon intensity APIs that provide real-time compute resources, storage operations, and network transfer content data. On-premises systems require direct power monitoring by intelligent PDUs or BMC telemetry, local grid carbon intensity data by a utility or other 3rd-party sources, like Electricity Maps or WattTime.

4.2 Carbon Budgets: Integrating Environmental Constraints into the Engineering Process

Following the pattern of latency SLAs and cost budgets, carbon budgets establish explicit constraints on the environmental impact of ML systems, integrating sustainability into engineering decision-making. A carbon budget specifies the maximum permissible emissions for a system over a defined period, creating accountability and forcing trade-off evaluation when proposed changes threaten to exceed allocations.

Implementing carbon budgets requires several components. First, baseline measurement establishes current emissions levels, providing the foundation for budget allocation. Historical analysis identifies emission trends, seasonal variations, and the relationship between business metrics—prediction volume, feature complexity—and carbon cost. Second, budget allocation distributes organizational carbon targets across teams and systems, analogous to departmental cost budgets or error rate objectives. A recommendation platform might receive an annual budget of 40 tons of CO2, subdivided into training (5 tons), inference (30 tons), and infrastructure (5 tons) allocations.

Carbon budgets introduce healthy tension into model development processes. Proposals to deploy marginally more accurate but substantially more computationally expensive models must justify the carbon cost increase. A model delivering 0.5% accuracy improvement while doubling inference latency and energy consumption faces scrutiny: Does the business value of marginal accuracy justify the environmental cost?

4.3 Architectural Optimization: Model Distillation, Geographic Placement, and Right-Sizing

Several architectural patterns consistently deliver carbon reductions across diverse ML applications. Model distillation represents one of the highest-leverage optimization techniques. By training compact models to approximate larger models' behavior, engineers can reduce inference costs by 60-80% while maintaining 90-95% of the original accuracy. Research by Alkhulaifi and colleagues examining knowledge distillation applications demonstrates that compression ratios between 4:1 and 50:1 can be achieved while retaining 92-98% of teacher model performance across computer vision and natural language processing domains [8]. The study documents that distilled models reduce inference latency by 40-85% and decrease memory requirements by 60-90% compared to original architectures, enabling deployment on resource-constrained devices while substantially lowering energy consumption.

10.48047/jocaaa.2026.35.03.12

Geographic compute placement maximizes infrastructure location for carbon intensity. Infrastructure optimization right-sizing removes unnecessary provision through over-provisioning and matches actual workload calculation with compute resources.

Infrastructure Metric	Value
GPU time per prediction (milliseconds)	100
Hardware power draw (watts)	250
Watt-hours per inference	0.007
Grid carbon intensity (grams CO2/kWh)	400
CO2 per prediction (grams)	0.003
Daily predictions (millions)	100
Annual CO2 emissions (tons)	110
Emissions reduction through metrics (%)	20-50
Compression ratio range (minimum)	4:1
Compression ratio range (maximum)	50:1

Table 2: Granular Carbon Measurement and Architectural Efficiency Gains [7,8]

5. The Accuracy-Carbon Trade-off

The quest to achieve advanced model precision has characterized the culture of ML engineering. Even relatively small gains in benchmark tests are celebrated by scientists and practitioners. But institutions and organizations need to explore the trade-off between the benefits of gaining accuracy and the carbon costs, since predictive systems are executing billions of inferences daily, and environmental concerns are escalating. Not all the improvements in performance are worth their environmental costs.

5.1 Diminishing Returns and the Cost of Marginal Gains

Model performance generally sees diminishing returns with increasing complexity. Early modeling attempts—moving from heuristics to simple ML, increasing feature sets, implementing more advanced algorithms—provide dramatic accuracy gains at relatively small computational expense. Subsequent stage optimizations seeking incremental improvements, however, tend to demand out-of-proportion resource expenditures. A study by Patterson and others that focused on the carbon footprint of large neural network training illustrates that reproducing state-of-the-art performance on benchmarking datasets demands models with heavily increased computational resources for providing only small accuracy gains [9]. The research finds that training a 213-million-parameter transformer model takes about 656,347 kWh of power, while marginal performance improvements can be sought by models using 10-100 times as many computational resources for improvements of just 2-5 percentage points. The recommendation accuracy may need model parameters to be doubled, training time to be tripled, and inference latency to be quadrupled to get the accuracy from 85% to 87%. The last percentage points of performance often require exponential scaling of resources.

This non-linearity presents an optimization problem. When computation resources seemed effectively unlimited and environmental externalities were unpriced, engineers reasonably sought any diminution in error rate at any cost. Carbon limits reverse this calculation, demanding clear consideration of whether marginal reductions in error rates yield value commensurate with environmental cost.

10.48047/jocaaa.2026.35.03.12

An example of a fraud-detection mechanism with an existing accuracy of 98.5 can be considered here. An offered model revision increases accuracy to 98.8%, preventing three extra cases of fraud per thousand transactions. Nevertheless, the update also takes up to 15,000ms. (rather than 5,000ms.) of inference latency, and triples the amount of memory used by GPS on the GPU. When the system capacity is sufficient to support 100 million transactions daily, the update adds an annual carbon emission of 40 and 120 tons of CO 2. Will the side-fixing of 80 more tons of emissions incur the marginalization of the reduction of 9,000 extra cases of fraud annually? It will hinge on the value of a given instance of fraud, the cost of false positives, and the priorities of the organization with respect to carbon.

5.2 Measuring Business Value In Terms of Environmental Cost

Accuracy-carbon trade-offs need to be assessed using frameworks that measure both variables in quantitative units. Engineers require methods of converting improvements in accuracy into business value metrics—revenue effect, cost savings, satisfaction of users, and carbon emissions into economic or ethical costs, including carbon prices, offset charges, and regulatory risk.

One solution sets up a shadow carbon price—a bookkeeping value per ton of CO2 emitted—so that cost-benefit analysis can be conducted in dollars. If an organization internally prices carbon at $100 per ton, the above fraud detection optimization costs an annualized carbon amount of $8,000. If avoiding 9,000 extra fraud instances saves $15,000 in value in terms of lowered losses and decreased operations cost, the optimization is of positive net worth. On the other hand, if the fraud case value merely averages $0.50, the $4,500 gain does not offset the carbon cost, even excluding infrastructure costs.

Other frameworks assess trade-offs based on operational measures. A recommendation algorithm may optimize for user activity measured in click-through rate. When a model that scores 3.2% CTR takes 50 tons CO2 per year and one that scores 3.3% CTR takes 85 tons, engineers can derive the carbon cost per basis point of CTR: 35 tons for 10 basis points, or 3.5 tons per basis point.

5.3 Sufficiency Thresholds and "Good Enough" Engineering

The idea of sufficiency—determining levels of performance above which increments add little value—constitutes a philosophical re-characterization of ML engineering goals. Studies on energy-aware machine learning models illustrate that energy-optimized design, instead of pure accuracy, can decrease power usage by 40-60% and still provide acceptable performance levels for production purposes [10]. The paper records that pruning methods, quantization strategies, and effective architecture choice allow for large reductions in energy consumption with little loss of accuracy. Instead of trying to maximize accuracy under computational constraints, engineers may try to minimize computation usage under sufficiency constraints on accuracy. Sufficiency constraints differ based on the application context. In life-critical applications such as medical diagnosis or autonomous driving perception, high accuracy needs may be worthy of high computational expenditure. However, for consumer applications like product recommendations, content personalization, or advertisement targeting, sufficiency thresholds emerge at lower accuracy levels.

10.48047/jocaaa.2026.35.03.12

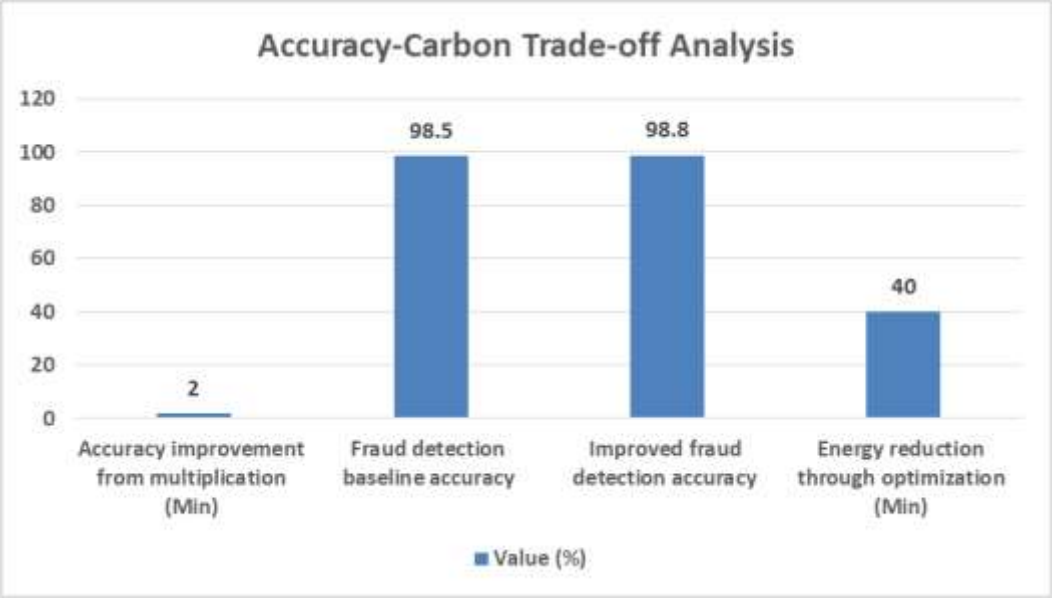

Figure 2: Accuracy-Carbon Trade-off Analysis [9,10]

Conclusion

The climate footprint of large-scale prediction systems is a salient but hidden externality that requires urgent engineering action. As machine learning infrastructures scale to billions of inferences per day, the aggregate carbon footprint shifts from de minimis to significant, necessitating systematic measurement, optimization, and accountability mechanisms. The structure introduced sets end-to-end methods for estimating carbon costs throughout training, inference, and infrastructure stages and illustrates that inference workloads often contribute the bulk of lifecycle emissions in spite of conventional emphasis on training expenses. Case studies confirm that significant reductions in emissions of thirty to forty percent are found to be feasible through model distillation, carbon-conscious load shifting, and optimization of infrastructure without sacrificing user-facing quality. The application of per-prediction carbon measurements, carbon budgets, and sufficiency thresholds allows platform engineers to make effective trade-offs between marginal improvements in accuracy and environmental expense. The intersection of environmental stewardship, economic efficiency, and technical excellence places carbon-aware engineering not as a constraint but as a driver of innovation. With the shift in regulatory pressure and demand requirements by the stakeholders, organisations that incorporate sustainability within the engineering of ML platforms will define competitive capabilities and favour global climate objectives. he shift from carbon-blind to carbon-aware ML engineering is coming regardless—the question is whether organizations choose to lead or follow.

References

[1] Emma Strubell et al., "Energy and Policy Considerations for Deep Learning in NLP", arXiv, 2019. [Online]. Available: https://arxiv.org/pdf/1906.02243

[2] Aniruddha Das and Avisikta Modak, "The Carbon footprint of Machine Learning Models", IJERA, 2023. [Online]. Available: https://ijera.in/index.php/IJERA/article/view/85/69

10.48047/jocaaa.2026.35.03.12

[3] Fernando Sevilla Martínez et al., "CO2 impact on convolutional network model training for autonomous driving through behavioral cloning", ScienceDirect, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034623000964

[4] Amir Moslemi et al., "A survey on knowledge distillation: Recent advancements", ScienceDirect, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827024000811

[5] Maraga Alex et al., "Carbon-Aware, Energy-Efficient, and SLA-Compliant Virtual Machine Placement in Cloud Data Centers Using Deep Q-Networks and Agglomerative Clustering", MDPI, Jul. 2025. [Online]. Available: https://www.mdpi.com/2073-431X/14/7/280

[6] OECD, "Measuring the Environmental Impacts of Artificial Intelligence Compute and Applications: The AI Footprint", 2022. [Online]. Available: https://www.oecd.org/content/dam/oecd/en/publications/reports/2022/11/measuring-the-environmental-impacts-of-artificial-intelligence-compute-and-applications_3dddded5/7babf571-en.pdf

[7] Abdolmaged Alkhulaifi et al., "Knowledge distillation in deep learning and its applications", National Library of Medicine, 2021. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8053015/

[8] David Patterson et al., "Carbon Emissions and Large Neural Network Training", arXiv. [Online]. Available: https://arxiv.org/pdf/2104.10350

[9] Rafał Różycki et al., "Energy-Aware Machine Learning Models—A Review of Recent Techniques and Perspectives", MDPI, May 2025. [Online]. Available: https://www.mdpi.com/1996-1073/18/11/2810