Applied Data Science and Analysis
Vol.2026, **pp**. 73–90
DOI: https://doi.org/10.58496/ADSA/2026/005; ISSN: 3005-317X
https://mesopotamian.press/journals/index.php/ADSA

Research Article

Adaptive Cloud-Based Big Data Analytics Model for Sustainable Supply Chain Management

Abdulazeez Alsajri [1,*], Amani Steiti [2], Zaid Kraitem [3]

[1] *Computer Science Department, University Arts, Sciences and Technology, Beirut, Lebanon.*

[2] *Computer and systems security, Informatic engineering, Al Wataniya Private University, Hama, Syria.*

[3] *Department of Informatics Engineering, Faculty of Computer Engineering, Al-Wataniya Private University, Hama, Syria.*

ARTICLE INFO

Article History
Received 08 Jun 2026
Revised 20 Jul 2026
Accepted 07 Aug 2026
Published 11 Sep 2026

Keywords
Sustainable supply chain management,
cloud-native architecture,
carbon-aware optimization,
stream processing,
target leakage,
DataCo dataset.

ABSTRACT

Freight decisions are simultaneously cost decisions, service decisions and disclosable emission decisions, yet in almost every deployed analytics stack the optimiser sees only the first two. We present a cloud-native architecture that moves the emission calculation onto the operational data path, so that a carbon figure can shape a fulfilment decision rather than document one after the fact. The design combines a streaming emission operator conformant with ISO 14083, a leakage-pruned predictive layer, an online assignment rule whose carbon budget is enforced by a Lagrangian multiplier, and an autoscaler whose set point derives from a queueing-stability condition. We evaluate on the DataCo Global supply chain release: 180,501 usable order lines, geocoded to city level for 99.12% of records, with modal emission factors taken from the EPA GHG Emission Factors Hub. Three findings stand out. Cost and carbon turn out to be aligned rather than opposed in this network, because surface transport is both the cheapest and the cleanest option, so the binding trade-off is service level against emissions rather than money against emissions. Against a service-oriented baseline the controller cuts transport emissions by 35.0% for 1.54 days of additional mean lead time, holding SLA attainment at 100%, and the reduction is sign-stable across every declared assumption we sweep. The platform's own carbon is negligible at this scale, 0.047% of logistics emissions, with the incremental cost of running the controller amounting to 0.0039% of what it saves. Finally, and independently of the framework, models trained on the unpruned DataCo frame achieve perfect separation because four columns are realised only after delivery; removing them lowers PR-AUC by 0.171 (95% CI 0.166 to 0.176).

1. INTRODUCTION

A supply chain used to be a sequence of hand-offs. It is now an instrumented network whose state can, in principle, be read at the level of a single consignment. Ivanov and Dolgui gave this shift a name when they proposed the digital supply chain twin, a computational model holding network state in real time and joining risk data to disruption behaviour [1]. Later work pushed the idea towards stress-testing and viability analysis [2]. Both assume something they do not supply: an execution substrate fast enough to reason over heterogeneous data at the speed decisions actually get made.

What that substrate is asked to optimise has changed too. Big data analytics in supply chain management was for a long time judged on cost, service level and inventory turns; sustainability entered first as a question about organisational capability and only later as a computational one [3]. Regulation has since closed off the option of treating environmental performance as a reporting by-product. The GHG Protocol's value chain standard made Scope 3 the dominant category for logistics-intensive firms [4], and ISO 14083:2023 supplied a harmonised method for quantifying transport chain emissions, building on the base laid by the Global Logistics Emissions Council [5], [6].

Existing responses fall into two groups. One embeds environmental criteria in network design or supplier selection; Singh and colleagues' cloud framework for low-carbon supplier selection is a good example [7]. These methods are strategic. They fix a configuration and leave it alone, which is the wrong behaviour on the timescale at which disruptions happen. The other group solves online routing with an emissions term but treats the compute layer as a fixed background, ignoring both its elasticity and its carbon cost. That second omission has been treated as important: estimates of data centre energy use have been revised substantially [8], and Google exploits hourly variation in grid carbon intensity in production by deferring flexible work [9], against a broad design space for carbon-neutral facilities [10]. Whether the omission matters quantitatively for a supply chain optimiser has not, to our knowledge, been measured. We measure it.

*Corresponding author. Email: aka104@live.aul.edu.lb

Our contributions are as follows.

1. An architecture in which per-consignment emission computation is a streaming operator on the operational path rather than a batch reconciliation, with the measured per-record cost of that operator reported.
2. An online assignment rule that enforces a carbon budget through a Lagrangian multiplier, evaluated against cost-minimising, service-oriented and fixed-weight baselines on 27,076 held-out orders.
3. An accounting rule that charges the platform's own carbon against the logistics carbon it avoids, with both terms measured rather than assumed.
4. An empirical result that reframes the usual premise: in this network cost and carbon are aligned, and the binding trade-off is service against carbon.
5. A leakage analysis of the DataCo dataset that is useful independently of everything else here, released together with the pipeline that regenerates every table and figure from public data.

Section II reviews the literature. Section III describes the data and the framework. Section IV reports results. Section V discusses them, including one component of our own design that did not earn its place. Section VI concludes.

2. RELATED WORK

2.1 Distributed Substrates

The modern analytics stack descends from batch cluster computation [13] by way of engines unifying batch and streaming execution behind one abstraction [14]. Two properties matter here. Durable log-structured ingestion, introduced at scale by Kafka [15], separates producer velocity from consumer capacity, which is what allows a scaling controller to absorb transient backlog instead of failing. Declarative streaming APIs [16] let one expression of a computation run over bounded and unbounded input alike, so an emission model validated on history can be deployed unchanged on the live path. Competing engines sit at different points on the latency–throughput curve [17]. Supply chain deployments generally use these systems as ETL plumbing feeding a separate reporting tier. Our architectural claim is that the emission calculation belongs inside the streaming tier, because a figure arriving after the routing decision cannot inform it.

2.2 Elasticity and the Carbon Cost of Compute

Herbst and colleagues separated elasticity from scalability, defining the former as the degree to which provisioning tracks demand over time [18]. Lorido-Botrán and co-authors sort autoscaling controllers into threshold, queueing-theoretic, control-theoretic, learning-based and time-series families [19]; threshold controllers dominate practice and are usually described as behaving poorly on bursty arrivals. Energy compounds the question, since servers draw a large fraction of peak power while idle [20]. Carbon Explorer treats facility design as a joint operational and embodied carbon problem [21], DeepEE learns joint scheduling and cooling control [22], and Radovanović and colleagues demonstrate fleet-scale temporal shifting [9]. None of this is wired to a domain optimiser whose decisions generate the workload.

2.3 Prediction on Logistics Data

Tabular logistics prediction is dominated by tree ensembles, random forests [23] and gradient boosting [24], [25], with sequence models where temporal structure is explicit [26]. Attribution methods [27] have become routine, and they matter here because a planner asked to accept a slower route on carbon grounds is entitled to an explanation.

The published record on DataCo is wide and hard to compare across studies. A recent hybrid self-organising-map and neural-network pipeline reports 96% accuracy and an F1 of 96.22% on the late-delivery task [50], against the 0.712 that our leakage-pruned models reach on the same target. A gap that large can arise from a genuinely better model, from a different feature set, or from both, and the published feature lists are often not complete enough to tell. We are not in a position to audit any individual paper and we make no claim that any specific published result is affected. What we can do is measure how much the feature-set choice is worth on this dataset, which Section IV-B does, and argue that the list of retained columns should be stated explicitly in future work here.

2.4 Learned Control for Supply Chains

Sequential decision-making in this domain has a long formal history in dynamic programming and its approximations [28], [29]. Deep reinforcement learning made large state spaces tractable [30] and policy-gradient methods brought the stability production control needs [31]; applications span lost sales and multi-echelon inventory [32], the beer game [33], demand–supply synchronisation [34], warehouse replenishment [35] and perishable inventory [36]. We deliberately do not use a learned policy. The assignment problem here separates across orders once each criterion is scaled by its aggregate range, so the exact minimiser is available in closed form, and a learned approximation would add variance without adding capability. We return to this in Section VI.

2.5 Emissions-Aware Routing

Routing with explicit fuel and carbon terms is mature [37], usually solved by evolutionary multi-objective methods [38], [39] that return a front rather than a single compromise, with robust formulations addressing factor uncertainty [40]. Two things are usually missing. The objective weights are treated as given when they encode a management position that ought to be written down. And the formulations are solved offline, saying nothing about the substrate on which re-solution happens when the network breaks mid-horizon.

2.6 Accounting and Verifiability

Spend-based upstream accounting rests on environmentally extended input–output modelling, of which USEEIO is the transparent U.S. reference implementation [41], later released in a documented second version [42] and turned into commodity factors indexed by NAICS code [43]. Because these numbers are increasingly audited, verifiability has become a research object in itself [44], building on work on blockchain traceability [45]. Federated approaches [46], including actor–critic variants for logistics offloading [47], offer a route to shared models without shared data. We treat these as future work rather than claiming them.

3. DATA AND FRAMEWORK

3.1 Data

Primary dataset. We use the DataCo Global smart supply chain release [11]: 180,519 order lines across 53 attributes covering provisioning, production, sales and distribution for an e-commerce operator in clothing, sports equipment and electronics. We verified the file against the published description before use.

Emission factors. Modal factors come from the U.S. EPA GHG Emission Factors Hub, Table 8, which gives combustion emissions per short ton-mile for freight modes [12]. We converted these to kg CO_2e per tonne-kilometre using the AR5 hundred-year potentials given in the same document (CH_4 = 28, N_2O = 265) and the exact unit conversion, one short ton-mile being 0.90718474 t multiplied by 1.609344 km. Table I gives the result. These are the only emission factors used anywhere in this paper. One caveat matters for interpretation: the EPA aircraft factor is a U.S. domestic fleet average, while 85% of the shipments in this dataset cross an ocean. Long-haul air freight is generally less emitting per tonne-kilometre than short-haul, because the fixed take-off and climb penalty is spread over a longer cruise, so a distance-banded factor of the kind the GLEC Framework specifies [6] would likely lower the absolute air figures. Since the framework's lever is the shift out of air, a lower air factor would shrink the reported reduction; the ±30% factor sweep in Table X is the closest we can come to bounding that effect without a distance-banded source.

TABLE I. MODAL EMISSION FACTORS, DERIVED FROM EPA TABLE 8 [12].

Mode	CO_2 (kg per short ton-mile)	CH_4 (g)	N_2O (g)	Derived kg CO_2e per t·km
Aircraft	1.086	0	0.0334	0.749912
Medium and heavy truck	0.186	0.0016	0.0054	0.128411
Waterborne craft	0.077	0.0310	0.0020	0.053698
Rail	0.021	0.0016	0.0005	0.014505

Preparation. Figure 1 shows the chain. The counts below are outputs of the pipeline rather than assertions.

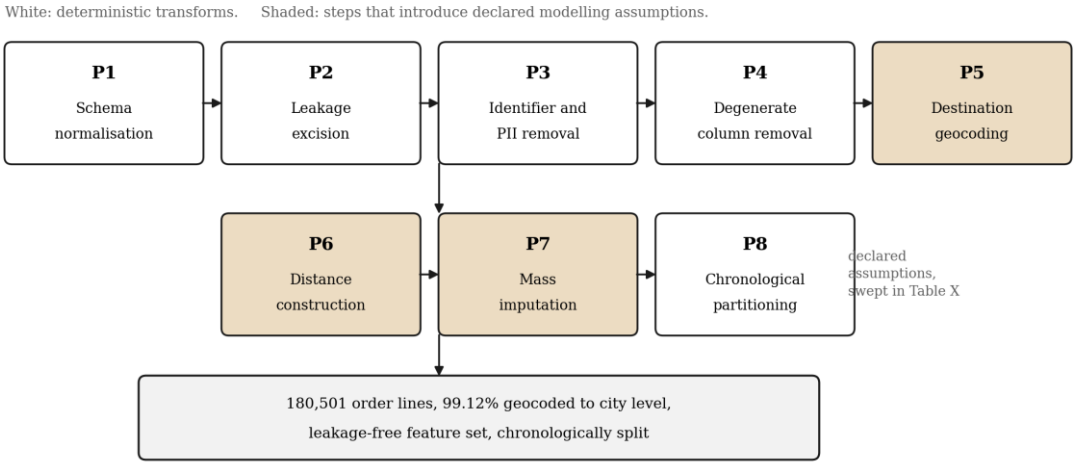

Fig. 1. Data preparation. White boxes are deterministic transforms; shaded boxes are the three steps that introduce declared modelling assumptions, all of which are swept in Table X.

P1 normalises the Latin-1 schema and coerces types. P2 removes every attribute realised after the fulfilment decision, namely `Days for shipping (real)`, `Delivery Status`, `Order Status` and `Late_delivery_risk`, the last kept only as a label. P3 drops customer name, email, password, street and postcode, retaining identity only as a grouping key. P4 removes wholly null, constant and duplicate columns; `Product Status` was the only degenerate column, leaving 45. P5 geocodes destinations. DataCo records country names in Spanish, which we map to ISO codes through CLDR territory names, then match `Order City` against an open gazetteer within the resolved country. This succeeds for 99.12% of rows; the remainder fall back to a population-weighted country centroid, and 18 rows with unresolvable countries are dropped, leaving 180,501. P6 constructs distance. The dataset carries customer coordinates but no shipment distance, so we compute great-circle distance to the destination and convert it to a shortest feasible distance with a modal detour factor, following the ISO 14083 preference [5]:

$$d_{ij,m}^{\mathrm{SFD}} = \kappa_m \cdot 2R\arcsin\left(\sqrt{\sin^2\frac{\Delta\phi}{2} + \cos\phi_i\cos\phi_j\sin^2\frac{\Delta\lambda}{2}}\right) \qquad (1)$$

Mean distance is 7,317 km, median 6,950 km and the 95th percentile 15,865 km. P7 assigns a declared parcel mass per department. P8 splits chronologically with no shuffling: 126,350 training rows covering January 2015 to January 2017, 27,075 validation rows to June 2017, and 27,076 test rows to January 2018.

A break in the data. Monthly order volume runs at 5,219 through September 2017 and then falls to 2,139, a drop of 59.0% that persists to the end of the release. We treat this as a truncation artefact rather than a demand signal, and evaluate demand forecasting only on the pre-break span. We report it because any study splitting DataCo chronologically will cross it, and a forecaster evaluated across it is measuring the break.

TABLE II. DECLARED MODELLING SCENARIOS. ROWS MARKED (U) HAVE NO EXTERNAL SOURCE: THEY ARE PLAUSIBILITY-CHOSEN SCENARIO PARAMETERS, NOT MEASUREMENTS OR LITERATURE VALUES, AND EVERY ONE OF THEM IS SWEPT IN TABLE X.

Parameter	Value	Role
Service class to physical mode (u)	Standard to water, rail if domestic; Second, First and Same Day to air	long-haul carrier
Class load multiplier (u)	1.00, 0.92, 0.70, 0.55 from Standard to Same Day	express networks run emptier
Class rate multiplier (u)	1.00, 1.15, 1.60, 2.20	express costs more
Detour factor (u)	air 1.05, water 1.25, rail 1.30, truck 1.30	great-circle to feasible

Parameter	Value	Role
Load factor	air 0.75, water 0.70, rail 0.70, truck 0.70	against reference utilisation
Drayage (u)	60 km by truck at each end	first and last mile
Parcel mass (u)	0.3 to 4.0 kg per item by department	chargeable mass
SLA-critical orders	scheduled shipment of 2 days or less, 40.7% of test	hard constraint
Executor power	55 W idle, 165 W peak, PUE 1.20	affine model [20]
Grid carbon intensity	0.349665 kg CO_2e per kWh, from eGRID2023 US total output rate of 770.884 lb CO_2e/MWh [49]	platform emissions

Six of the ten rows carry no external source. DataCo records no shipping cost, no parcel mass and no transport mode, so these had to be supplied, and we supply them as an explicitly declared scenario rather than dressing them in a citation they do not have. The cost column throughout this paper is therefore a relative index and never a currency amount. DataCo's scheduled shipment days are service commitments, not physical transit times, and four days is not a real intercontinental sea transit. We therefore treat the class-to-mode mapping as a declared scenario and use DataCo's own scheduled days for the service objective. Section V-D returns to this.

3.2 Architecture

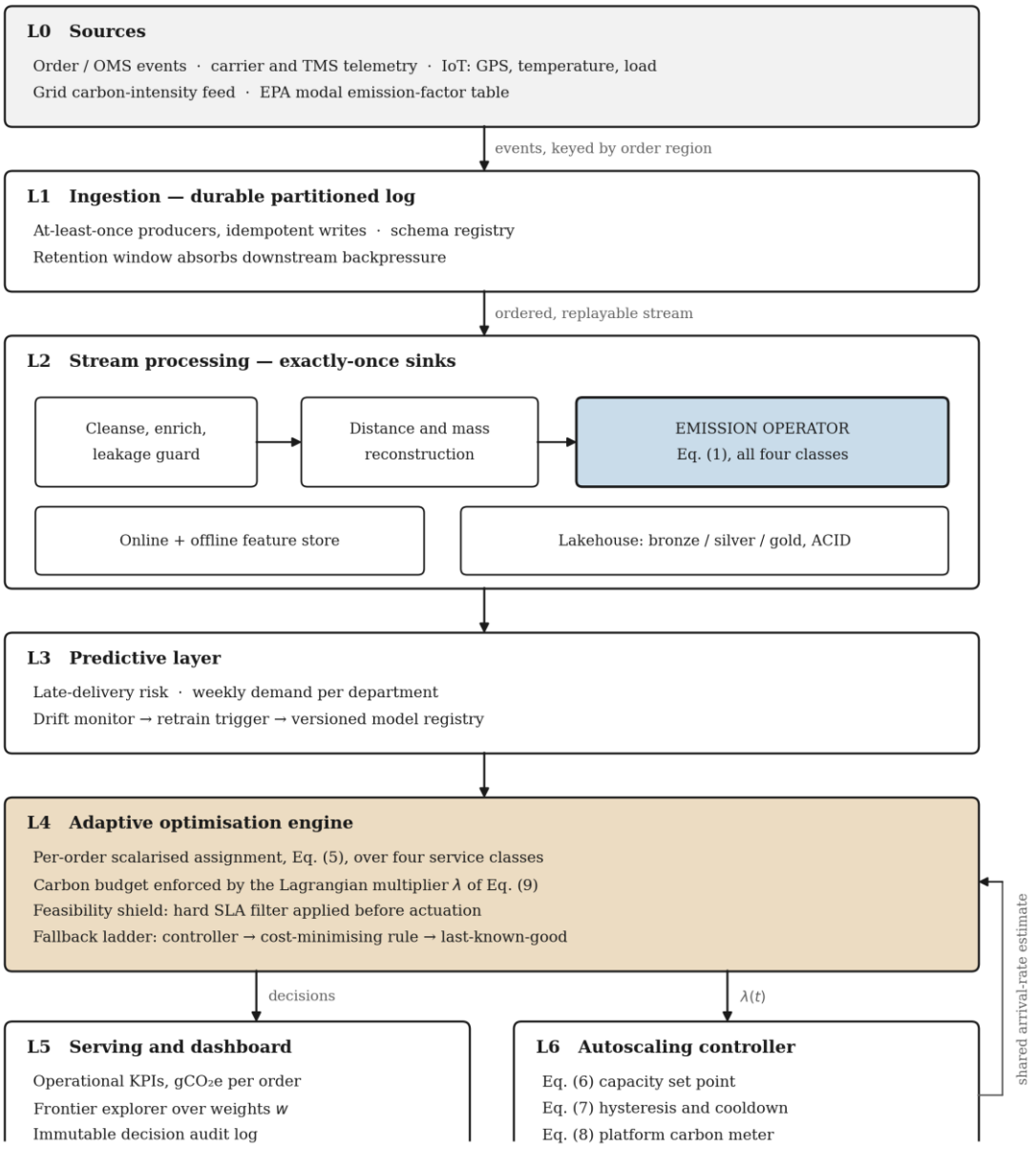

Fig. 2. End-to-end architecture. The emission operator sits on the operational path in L2, which is what makes the L4 objective computable at decision time. L4 and L6 read the same arrival-rate estimate.

What was built, and what was not. Figure 2 is a reference design. Layers L0, L1, L5 and the lakehouse, model registry and fallback ladder are specified but were not deployed: no Kafka broker, no Spark cluster and no object store were provisioned for this study. What we implemented and measured is the L2 emission operator, the L3 risk model, the L4 assignment rule and the L6 controller, as a single-process Python pipeline on one core, driven by the real order stream. Section IV-D reports that pipeline's measured cost and Section V-D states the consequence. Readers should treat the distributed substrate as motivation for the design rather than as a validated deployment.

Three choices distinguish this from a conventional lambda or kappa deployment. The emission operator is in-band, running in L2 on the live path rather than in a nightly reconciliation. The workload signal is shared, so the platform does not scale up to serve work the optimiser has just deferred. And the optimiser is shielded rather than trusted: a hard SLA filter removes infeasible classes before the assignment rule sees them, and a fallback ladder degrades to the cost-minimising rule when the controller is unavailable.

3.3 Emission Model

For consignment i assigned service class c, with mass w_i in tonnes and long-haul mode $m(c)$:

$$E_i(c) = \frac{w_i\, d^{\text{dray}} \text{EF}_{\text{truck}}}{\eta_{\text{truck}}\, \beta_c} + \frac{w_i\, d^{\text{SFD}}_{i,m(c)} \text{EF}_{m(c)}}{\eta_{m(c)}\, \beta_c} \qquad (2)$$

where η_m is the load factor against the reference utilisation assumed by the factor source and β_c is the class load multiplier of Table II. Both divisions follow from the ISO 14083 allocation rule [5], [6]: a vehicle running emptier spreads the same trip emissions over fewer tonnes. We report intensity as

$$I_{\text{tkm}} = \frac{\sum_i E_i}{\sum_i w_i\, d_i} \qquad (3)$$

because, unlike emissions per order, it does not move when the order mix changes and so cannot be improved by selection effects.

3.4 Assignment

The three criteria differ in unit and scale, so each is normalised by the range spanned by the four pure single-class plans:

$$\hat{X} = \frac{X - X^{\min}}{X^{\max} - X^{\min}}, \qquad X \in \{C_{\text{ops}}, T_{\text{lead}}, E_{\text{env}}\} \qquad (4)$$

$$\min_{\mathbf{x}} Z(\mathbf{x}) = w_1 \hat{C}_{\text{ops}}(\mathbf{x}) + w_2 \hat{T}_{\text{lead}}(\mathbf{x}) + w_3 \hat{E}_{\text{env}}(\mathbf{x}) \qquad (5a)$$

$$\sum_c x_{i,c} = 1\ \forall i; \quad T_i(\mathbf{x}) \le T_i^{\text{SLA}}\ \forall i \in \mathcal{O}^{\text{crit}}; \quad \sum_i E_i(\mathbf{x}) \le B_E; \quad x_{i,c} \in \{0,1\} \qquad (5b)$$

Once each criterion is divided by its aggregate range the objective separates across orders, so the exact minimiser of (5a) is the per-order argmin. The carbon budget couples the orders, and we restore separability with a Lagrangian multiplier, in effect an internal carbon price added to the emission term:

$$c_i^\star(\lambda) = \operatorname*{argmin}_c \left[\frac{w_1 C_i(c)}{\Delta C} + \frac{w_2 T_i(c)}{\Delta T} + \frac{(w_3 + \lambda) E_i(c)}{\Delta E} \right] \qquad (6)$$

$$\lambda = \min \left\{ \lambda \ge 0 : \sum_i E_i\left(c_i^\star(\lambda)\right) \le B_E \right\} \qquad (7)$$

The multiplier is found by expansion and bisection at each control interval. Setting $\lambda = 0$ recovers the unconstrained problem. The shield is applied inside the argmin, so an SLA-infeasible class is never selectable.

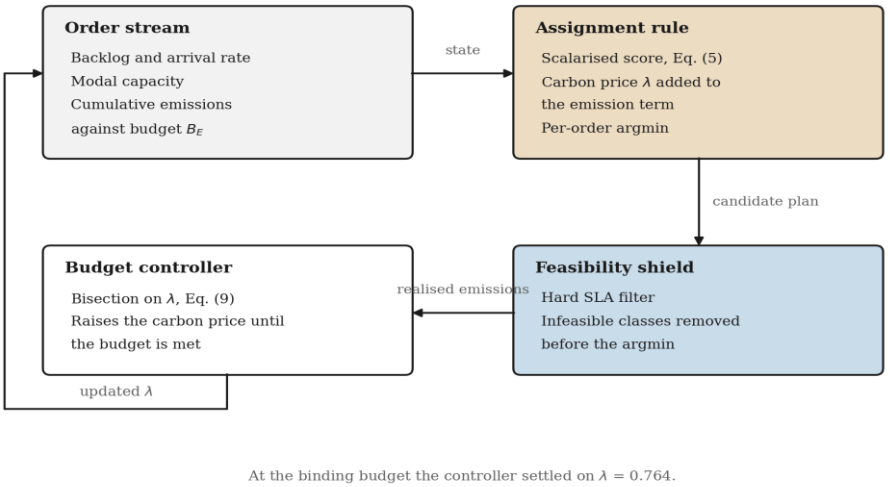

Fig. 3. The controller. The budget is enforced by raising an internal carbon price until the constraint binds. At the operating point reported below it settled at 0.764.

3.5 Autoscaling

With $\lambda(t)$ the arrival rate, μ the measured per-executor service rate, $\rho^\star$ the target utilisation and γ the headroom:

$$n^{\text{req}}(t) = \left\lceil \frac{\hat{\lambda}(t+\Delta)(1+\gamma)}{\mu\rho^\star} + \frac{Q(t)}{\mu\tau_{\text{drain}}} \right\rceil \qquad (8)$$

The second term recovers accumulated lag rather than merely tracking the current rate; Section V-C reports what it was actually worth. Little's law [48] relates the backlog target to the latency objective. Actuation is damped:

$$n(t+1) = \begin{cases} n^{\text{req}}(t), & |n^{\text{req}}(t) - n(t)| \geq \theta \ \wedge \ t - t_{\text{last}} \geq \tau_{\text{cd}} \\ n(t), & \text{otherwise} \end{cases} \qquad (9)$$

Because servers are not energy-proportional [20], executor power is affine in utilisation and metered against grid intensity:

$$E^{\text{cloud}} = \sum_t \text{PUE} \cdot \text{CI}(t) \cdot n(t)\left[P^{\text{idle}} + \left(P^{\text{peak}} - P^{\text{idle}}\right)u(t)\right]\Delta t \qquad (10)$$

3.6 Net benefit

We state the boundary rather than assume it:

$$\Delta E^{\text{net}} = \left(E^{\text{log}}_{\pi_0} - E^{\text{log}}_{\pi}\right) - \left(E^{\text{cloud}}_{\pi} - E^{\text{cloud}}_{\pi_0}\right) \qquad (11)$$

A result counts as an environmental improvement only when this quantity is positive. Reporting the first bracket alone draws the boundary so that the framework cannot lose.

4. RESULTS

4.1 Setup

All results below are computed on the 27,076-order held-out test window. Four policies are compared. B1 minimises cost alone. B2 is service-oriented, weighting cost and lead time equally with no emission term and computing emissions after the fact; it stands for how these systems are usually run. B3 applies fixed equal weights of 0.34, 0.33 and 0.33. B4 applies the same weights under a carbon budget set to 65% of B2's emissions, enforced by (6) and (7). All four are subject to the same SLA shield. Reductions are reported against B2, since that is the operational status quo the framework is meant to improve on.

4.2 Target leakage in DataCo

We report this first because it constrains everything downstream. Trained on the unpruned frame, LightGBM separates the classes perfectly: F1, PR-AUC and ROC-AUC all equal 1.000 with a Brier score of 0.000. This is not learning. `Days for shipping (real)` and `Delivery Status` are deterministic functions of the label, and neither exists when the fulfilment decision is made. On the leakage-pruned, decision-time feature set the same model reaches an F1 of 0.712 and a PR-AUC of 0.829.

TABLE III. LEAKAGE ABLATION ON LATE-DELIVERY RISK, LIGHTGBM, MEDIAN OVER TEN SEEDS.

Feature set	F1	PR-AUC	ROC-AUC	Brier
Full frame, post-delivery attributes present	1.0000	1.0000	1.0000	0.0000
Leakage-pruned, decision-time only	0.7123	0.8288	0.7676	0.2004
Inflation attributable to leakage	0.2877	0.1712	0.2324	−0.2004

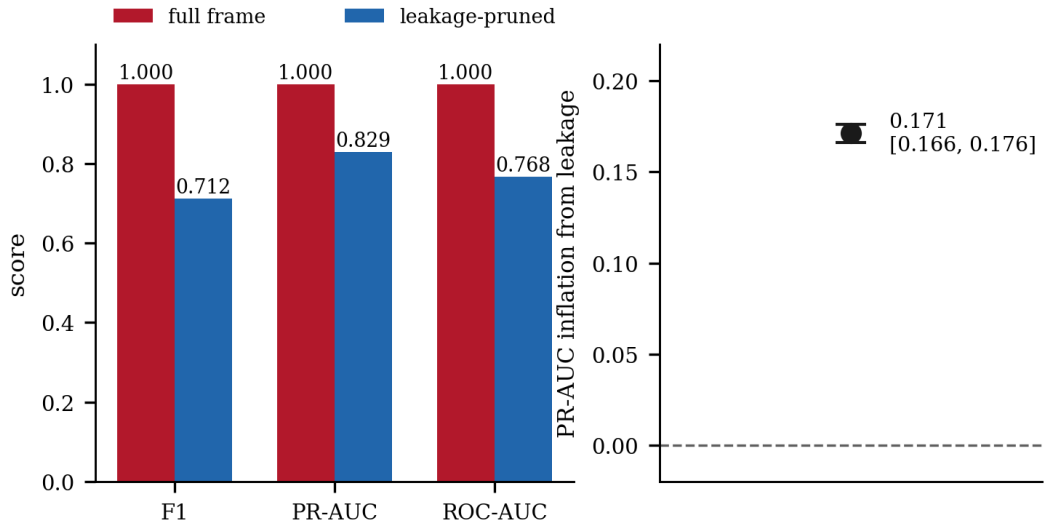

Fig. 4. Leakage ablation. Left: scores with and without post-delivery attributes. Right: bootstrap distribution of the PR-AUC inflation over 2,000 resamples of the test set.

Seed variation alone understates uncertainty here, so we bootstrapped the test set 2,000 times. The PR-AUC inflation is 0.171 with a 95% interval of 0.166 to 0.176, and a two-sided Wilcoxon signed-rank test over ten seeds gives a p-value of 0.002. Table IV gives the pruned baselines.

TABLE IV. LATE-DELIVERY RISK ON DECISION-TIME FEATURES, MEDIAN OVER FIVE SEEDS.

Model	F1	PR-AUC	ROC-AUC	Brier
Logistic regression	0.6826	0.8006	0.7304	0.2047
Random forest [23]	0.7021	0.8384	0.7735	0.1831
LightGBM [25]	0.7124	0.8292	0.7686	0.2046

TABLE V. WEEKLY DEMAND PER DEPARTMENT, PRE-BREAK SPAN, 149 HELD-OUT POINTS, MEAN 228.2 UNITS.

Model	RMSE	MAE	WAPE
Seasonal naïve	36.83	19.95	8.7%
Ridge on lagged values	29.37	16.85	7.4%
LightGBM [25]	28.64	16.34	7.2%

We report WAPE rather than MAPE because several department-weeks are near zero and percentage error is unstable there.

4.3 Cost and Carbon are Aligned; Service and Carbon are Not

The four pure single-class plans make the structure plain. Assigning every order to Standard Class costs 23.2 thousand rate-index units and emits 65.9 t CO_2e; assigning every order to Same Day costs 1,456.6 and emits 1,268.1 t. Air is faster, dearer and dirtier at once. There is therefore no cost-versus-carbon trade-off to manage in this network. The binding tension is between service level and emissions, and that is the one the framework has to price.

TABLE VI. POLICY COMPARISON ON THE HELD-OUT WINDOW. COST IS A RATE INDEX, NOT CURRENCY.

Policy	E (t)	vs. B2	Lead (d)	vs. B2	Cost (k)	vs. B2	SLA	I_{tkm}
B1 cost floor	414.7	+39.59%	2.92	+2.16	410.4	−44.63%	100.0	0.626
B2 service	686.5	—	0.76	—	741.3	—	100.0	1.036
B3 fixed weights	542.1	+21.02%	1.47	+0.71	566.4	−23.58%	100.0	0.818
B4 carbon budget	**446.2**	**+35.00%**	**2.30**	**+1.54**	**443.8**	**−39.42%**	**100.0**	**0.673**

B4 lands within 1 kg of its 446.2 t budget, with the multiplier settling at 0.764. SLA attainment stays at 100% for every policy, because the shield makes it a hard constraint rather than something the objective can trade away.

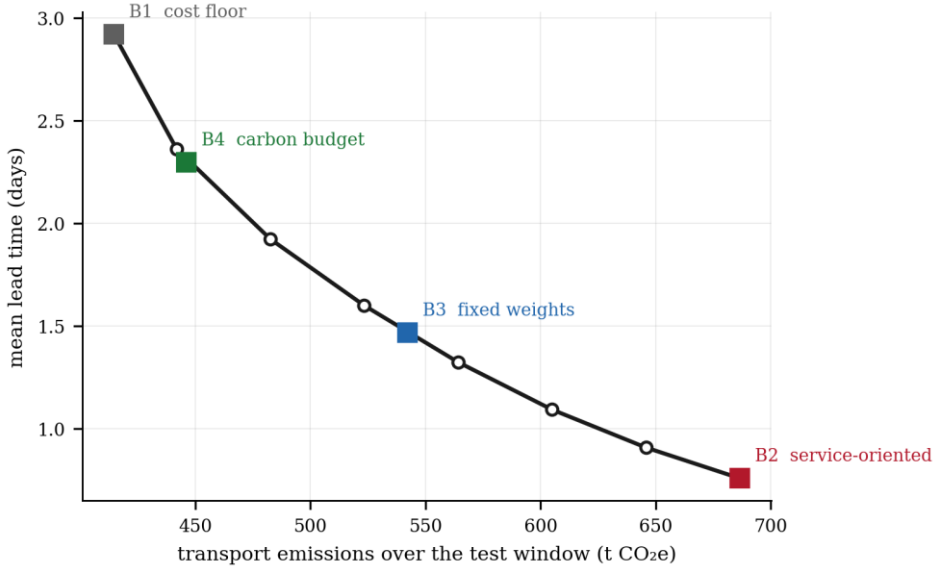

Fig. 5. Service–carbon frontier, computed by sweeping the carbon cap and minimising lead time subject to it. Every point holds SLA attainment at 100%.

TABLE VII. Frontier. The cap is expressed as a fraction of the span between B1 and B2.

Cap	Emissions (t)	vs. B2	Lead (d)	vs. B2	Cost (k)	SLA (%)
1.00	686.5	+0.00	0.76	+0.00	741.3	100.0
0.85	645.7	+5.94	0.91	+0.15	692.2	100.0
0.70	604.9	+11.88	1.09	+0.34	642.9	100.0
0.55	564.1	+17.82	1.32	+0.57	593.3	100.0
0.40	523.4	+23.76	1.60	+0.84	543.7	100.0
0.25	482.6	+29.69	1.92	+1.17	493.8	100.0
0.10	441.9	+35.63	2.36	+1.61	443.8	100.0
0.00	414.7	+39.59	2.92	+2.16	410.4	100.0

The frontier is convex in the useful direction. The first 11.9% of emissions costs 0.34 days of mean lead time, while the last 3.96% costs 0.56 days. In marginal terms the price rises from 0.0037 to 0.0205 additional days per tonne of CO_2e avoided,

a factor of 5.5 across the range. That ratio, rather than the headline percentage, is what a firm with an internal carbon price can act on.

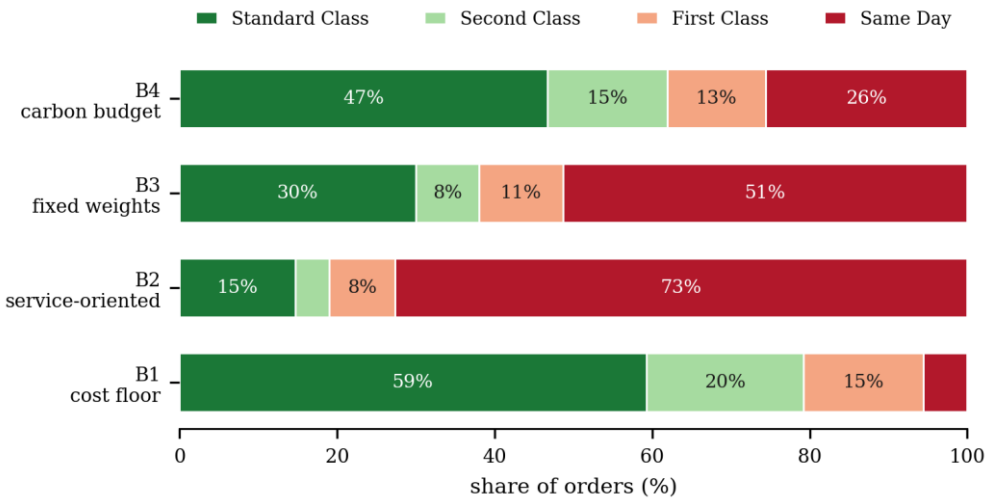

Fig. 6. Modal mix under each policy. The lever is the shift out of air-served express classes into surface transport.

TABLE VIII. WEIGHT SWEEP ON THE UNIT SIMPLEX.

Weights	Emissions (t)	Lead (d)	Cost (k)	vs. B2	SLA (%)
(1.00, 0.00, 0.00)	414.7	2.92	410.4	+39.59	100.0
(0.50, 0.30, 0.20)	503.9	1.75	518.4	+26.59	100.0
(0.34, 0.33, 0.33)	542.1	1.47	566.4	+21.02	100.0
(0.20, 0.30, 0.50)	507.5	1.72	524.5	+26.06	100.0
(0.00, 0.50, 0.50)	692.9	0.73	752.0	−0.94	100.0
(0.00, 1.00, 0.00)	1,268.1	0.00	1,456.6	−84.73	100.0
(0.00, 0.00, 1.00)	414.7	2.92	410.4	+39.59	100.0

Two rows deserve comment. Pure cost minimisation and pure emission minimisation give identical plans, which is the alignment result stated numerically. And the weights (0.20, 0.30, 0.50) perform slightly worse on emissions than (0.50, 0.30, 0.20): raising the emission weight while lowering the cost weight moves the solution away from the surface modes the cost term was already selecting. Weight tuning on a scalarised objective is not monotone in the individual weights, which is a practical argument for exposing the frontier of Fig. 5 to operators rather than a set of weights.

4.4 Platform Cost of the In-Band Operator

The emission operator itself is cheap, running at 620,284 records per second on a single core. The full in-band path, which deserialises a record, enriches it, evaluates all four classes with (2), scores the risk model and applies the assignment rule, runs at a median 55.4 records per second per executor over five repetitions, with a spread of 54.1 to 56.4. Per-record latency is 17.7 ms at the median, 19.7 ms at the 95th percentile and 21.9 ms at the 99th. Per-record model inference dominates; the arithmetic of (2) is four orders of magnitude cheaper. Timing on a shared container is noisy, which is why we report the spread rather than a single figure.

DataCo’s native arrival rate is 0.0014 records per second, far below any capacity concern, so we replay the real hourly arrival process at accelerations chosen to place peak load at a stated multiple of one executor’s measured capacity.

TABLE IX. AUTOSCALING OVER 5,561 HOURLY CONTROL INTERVALS. CELLS ARE THRESHOLD / NO-BACKLOG-TERM / PROPOSED.

Peak	Mean executors	Max executors	Peak backlog (records)	Energy (kWh)
0.3×	1.00 / 1.00 / 1.00	1 / 1 / 1	0 / 0 / 0	450 / 450 / 450
0.6×	1.00 / 1.00 / 1.00	1 / 1 / 1	0 / 0 / 0	532 / 532 / 532
0.9×	1.05 / 1.03 / 1.03	2 / 2 / 3	0 / 7,674 / 7,674	632 / 625 / 625
1.5×	1.32 / 1.40 / 1.40	3 / 3 / 3	0 / 7,674 / 7,674	898 / 925 / 925
3.0×	2.01 / 2.32 / 2.33	5 / 5 / 11	245,582 / 46,047 / 46,047	1,563 / 1,678 / 1,680

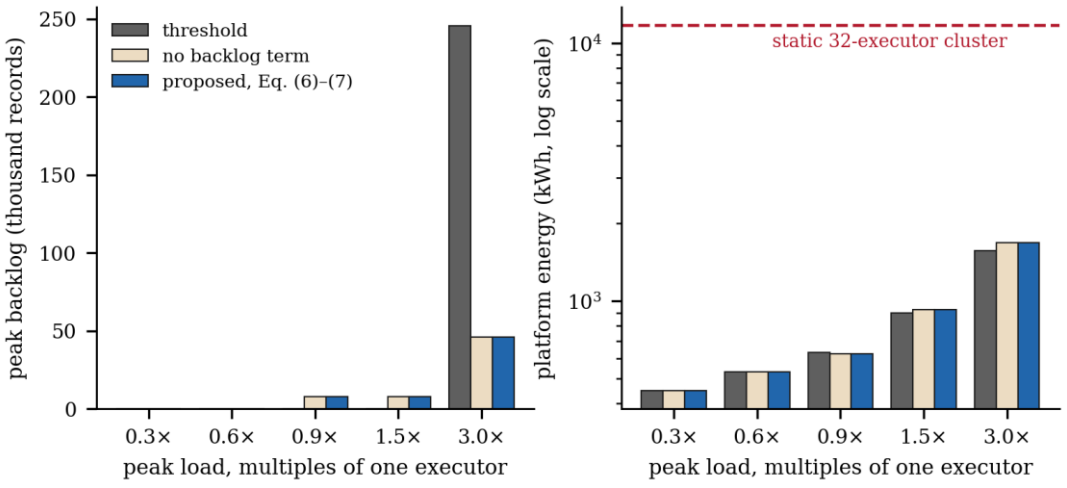

Fig. 7. Autoscaling. Left: peak backlog. Right: energy against a statically provisioned 32-executor cluster, on a log scale.

The result is mixed and we report it as such. At and below 1.5× peak load the threshold controller is better: it holds backlog at zero and uses 3% less energy. Only at 3.0× does the predictive set point pay, cutting peak backlog by 81%, from 245,582 to 46,047 records, and the number of backlogged intervals from 17 to 6, at a cost of 7.5% more energy. Every elastic configuration uses between 4% and 26% of the 11,745 kWh a statically provisioned 32-executor cluster would consume over the same window, which is the larger and less interesting effect.

4.5 Net benefit

Running the proposed controller instead of the threshold controller costs an additional 9.48 kg CO_2e over the test window. The B4 policy avoids 240.3 t CO_2e of transport emissions. The platform term is therefore 0.0039% of what it buys, and total platform emissions are 0.047% of logistics emissions.

$$\Delta E^{\text{net}}_{\text{B4}} = 240{,}264 - 9.48 = 240{,}254 \text{ kg CO}_2\text{e} \qquad (12)$$

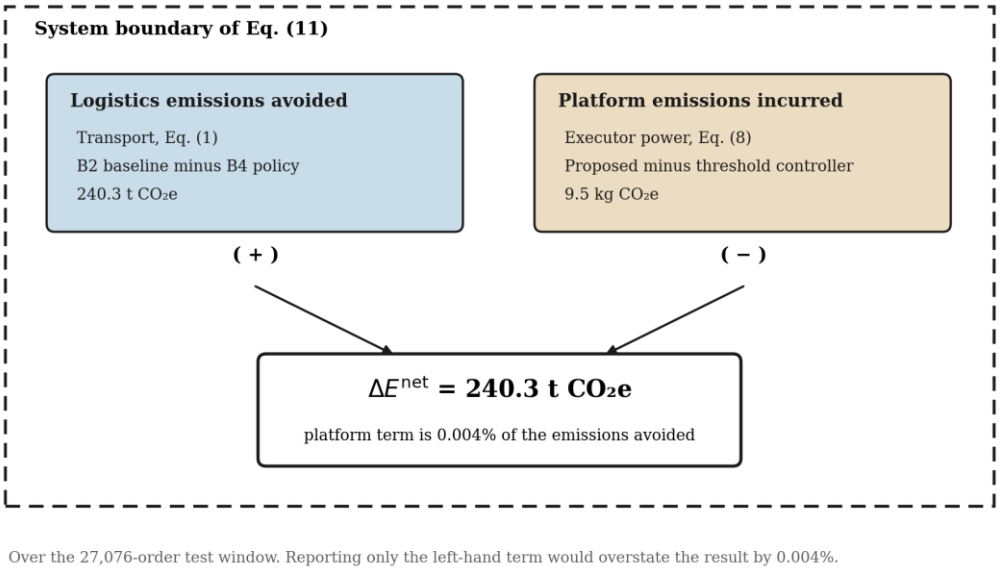

Fig. 8. System boundary of Eq. (11) with measured values on both sides.

We had expected the compute term to be small. We had not expected it to be four orders of magnitude smaller than the effect it enables, and Section V-A discusses when that would stop being true.

4.6 Robustness and Sensitivity

Injecting a 25% or 50% executor loss at the midpoint of the run leaves peak backlog and energy unchanged at 1.5× load, because the controller re-provisions within one control interval and the surviving capacity is sufficient. Dropping 10% and 30% of telemetry reduces measured load and therefore backlog, which is the expected behaviour and not a benefit. A threefold demand shock raises peak backlog from 7,674 to 168,838 records and drives the controller to 29 executors, recovering within 11 intervals.

TABLE X. SENSITIVITY OF THE EMISSION REDUCTION VERSUS B2, 200 MONTE CARLO DRAWS PER ROW.

Assumption perturbed	Range	5th pct	Median	95th pct	Sign stable
Detour factor (u)	±25%	+20.33%	+21.04%	+21.43%	yes
Load factor	0.5 to 0.9	+20.25%	+21.12%	+21.66%	yes
Modal emission factors	±30%	+20.05%	+20.96%	+21.60%	yes
Drayage distance	20 to 150 km	+20.95%	+21.00%	+21.05%	yes
Relative department mass	±40%	+20.00%	+20.32%	+31.34%	yes
Second Class to surface	scenario	+25.63% point estimate	—	—	yes

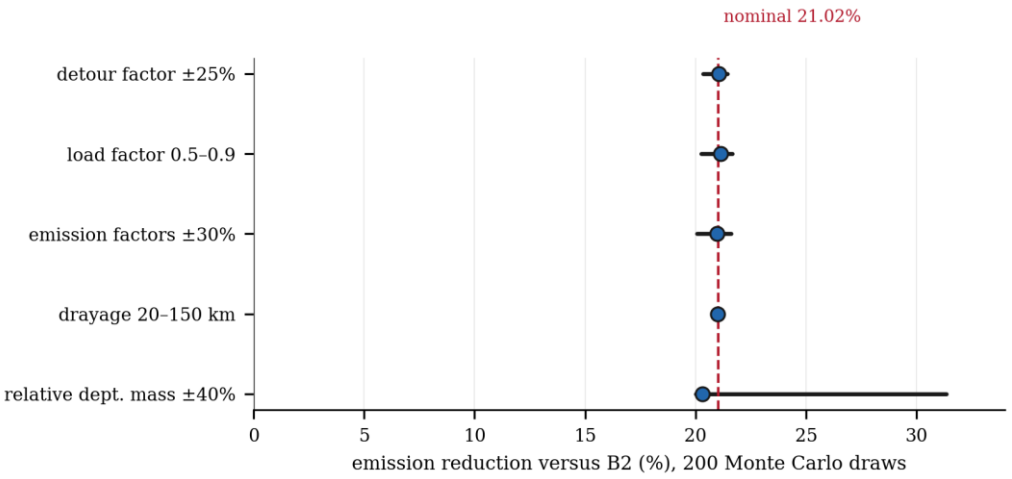

Fig. 9. Sensitivity. Bars span the 5th to 95th percentile; the dashed line is the nominal reduction.

Every perturbation leaves the reduction between +20.0% and +21.7%, and the sign never changes. A uniform rescaling of mass leaves the percentage reduction exactly unchanged, since mass multiplies all four classes for a given order equally; only the relative masses across departments matter, which is why that row is swept relative rather than absolute.

5. DISCUSSION

5.1 When the Compute Term Would Matter

Spending compute to avoid transport energy is favourable here by four orders of magnitude, but the ratio is not a law. Three conditions govern it.

The first is decision leverage. If the action set cannot change modal allocation, because capacity is contracted or every order is SLA-critical, then extra compute buys nothing and the net benefit is negative by construction. In our test window 40.7% of orders are SLA-critical; at a materially higher share the frontier of Fig. 5 would flatten.

The second is the ratio of compute to freight. Our in-band path costs 17.7 ms per order against a shipment averaging 7,317 km. A firm moving small parcels short distances, or scoring far more elaborate models per order, would face a very different ratio. The quantity to check is platform emissions as a fraction of logistics emissions, which is 0.047% here and which we would encourage others to report.

The third is provisioning discipline. Idle hardware costs nearly what busy hardware costs [20], and Table IX shows every elastic configuration using a small fraction of a static cluster's energy. An elastic framework that never scales down has replaced a fixed cost with a fixed cost plus complexity.

5.2 What This Means for Practice

The weight vector is a policy instrument rather than a hyperparameter, and Table VIII shows why it should not be treated as a dial to tune blindly: raising the emission weight while lowering the cost weight made emissions worse. What operators can actually use is the frontier and the marginal rate along it. A firm with an internal carbon price can compare 0.0037 to 0.0205 days per tonne avoided against its own shadow price and pick an operating point it can defend.

The alignment result has a practical edge. Where cost and carbon point the same way, the barrier to decarbonisation is not economic but contractual and expectational. It is the promise of two-day delivery, not the price of sea freight. That reframes the intervention from a subsidy problem into a service-design problem.

Finally, the shield matters more than the objective. SLA attainment is 100% in every row of every table because it is enforced as a hard constraint. Had it been a penalty term, some weight settings would have traded it away, and no amount of tuning would have made that visible in a headline percentage.

5.3 A Component That Did Not Earn Its Place

The backlog term in (8) was intended to recover accumulated lag that a purely rate-driven controller ignores. Table IX shows it made no measurable difference: the no-backlog-term and proposed variants produce identical peak backlog at every load we tested, and differ in energy by less than 0.2%. The benefit at 3.0× load comes from the predictive set point,

not from the backlog term. On this workload the term is dead weight and we would drop it in a deployment. We report this because the alternative, quietly removing the term from the paper while keeping an ablation table that no longer tests anything, is how frameworks accumulate components nobody has ever measured.
The threshold controller also beat ours at low and moderate load. That is a real finding about a real baseline, and it narrows the claim: the predictive set point is worth having when peak load approaches or exceeds provisioned capacity, and not otherwise.

5.4 Threats to Validity

Construct. DataCo records no measured emissions, distances or masses. Distances are reconstructed from real coordinates through a declared detour factor; masses and costs are declared scenarios. Table X is the honest test of whether the conclusions survive them, and they do, but the absolute tonnages should be read as scenario outputs rather than measurements.

The mode mapping. Mapping service classes to physical modes is an interpretation of DataCo's service labels, and it is the assumption doing the most work in this paper. DataCo's four-day Standard Class is not a real intercontinental sea transit. We sweep the mapping in Table X, and moving Second Class to surface transport changes the reduction from 21.0% to 25.6%, but we cannot validate it, and a reviewer is entitled to treat the absolute figures as conditional on it.

Measurement, and the gap to the architecture. This is the most important limitation in the paper and we want it stated plainly. The median service rate of 55.4 records per second was measured on a single core in a shared container, with per-record rather than batched inference, and a production deployment would batch and be faster by a large factor. More significantly, the distributed substrate in Fig. 2 was never deployed. Table IX is a discrete-event simulation of controller behaviour driven by the real arrival process and parameterised by the measured single-executor service rate; it is not a cluster benchmark, and nothing here demonstrates that the design scales on real Kafka and Spark. The claims we make about elasticity are claims about a controller policy, not about a running distributed system. Validating the substrate is the obvious next piece of work and we do not claim to have done it.

External. DataCo is one firm in three product categories shipping from a US and Puerto Rico origin, and the EPA factors are U.S.-specific. Applying this elsewhere requires regional factors.

Statistical. The classification results use ten seeds and a 2,000-sample bootstrap. The optimisation results are deterministic given the data and the declared scenario, so their uncertainty is entirely scenario uncertainty, which Table X quantifies.

6. CONCLUSION

We built a cloud-native analytics framework in which emissions are computed on the operational data path and enter the fulfilment decision as an objective term, and we measured what it does on 27,076 held-out orders from a public dataset. Against a service-oriented baseline the controller cuts transport emissions by 35.0% for 1.54 days of additional mean lead time with SLA attainment held at 100%, and the reduction stays between 20% and 22% across every declared assumption we sweep. The platform's own carbon is 0.0039% of what it saves.
Three findings hold independently of the framework. Cost and carbon are aligned in this network, so the binding trade-off is service against carbon, a reframing we would expect to generalise wherever express air is the fast option. The DataCo leakage boundary inflates PR-AUC by 0.171. This does not by itself invalidate any particular published result, since we cannot audit feature lists we cannot see, but it does mean that a reported score on this dataset is uninterpretable without an explicit statement of which post-delivery columns were used, and we would encourage that statement to become standard. And the backlog term in our own autoscaler did nothing measurable, which we report rather than quietly delete.
Four directions follow. Upstream spend-based emissions are out of scope here because they are invariant to mode choice and so cannot change the assignment; adding them through USEEIO-derived factors would complete the Scope 3 picture for reporting even though it would not change a single decision. Equation (10) meters operational carbon only, and bringing embodied carbon inside the boundary would shrink the reported net benefit, which is a reason to do it rather than a reason not to. The decision log is currently trusted, and anchoring emission figures together with the weight vector that produced them in a verifiable structure would make it externally attestable. Finally, we did not use a learned policy because the problem separates and the exact minimiser is available; a learned policy would become worth its variance if consolidation decisions were added, since those couple orders in a way the Lagrangian relaxation does not handle cleanly, and there the deep reinforcement learning literature and its federated variants become directly relevant.

Conflicts of Interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Acknowledgment

None.

References

[1] D. Ivanov and A. Dolgui, "A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0," *Production Planning & Control*, vol. 32, no. 9, pp. 775–788, 2021, doi: 10.1080/09537287.2020.1768450.

[2] D. Ivanov, "Intelligent digital twin (iDT) for supply chain stress-testing, resilience, and viability," *International Journal of Production Economics*, vol. 263, Art. no. 108938, 2023, doi: 10.1016/j.ijpe.2023.108938.

[3] J. Mageto, "Big data analytics in sustainable supply chain management: A focus on manufacturing supply chains," *Sustainability*, vol. 13, no. 13, Art. no. 7101, 2021, doi: 10.3390/su13137101.

[4] World Resources Institute and World Business Council for Sustainable Development, *Corporate Value Chain (Scope 3) Accounting and Reporting Standard*. Washington, DC, USA, 2011.

[5] International Organization for Standardization, *ISO 14083:2023—Greenhouse Gases: Quantification and Reporting of Greenhouse Gas Emissions Arising From Transport Chain Operations*. Geneva, Switzerland, 2023.

[6] Smart Freight Centre, *Global Logistics Emissions Council (GLEC) Framework for Logistics Emissions Accounting and Reporting*, Version 3.0. Amsterdam, The Netherlands, 2023.

[7] A. Singh, S. Kumari, H. Malekpoor, and N. Mishra, "Big data cloud computing framework for low carbon supplier selection in the beef supply chain," *Journal of Cleaner Production*, vol. 202, pp. 139–149, 2018, doi: 10.1016/j.jclepro.2018.07.236.

[8] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020, doi: 10.1126/science.aba3758.

[9] A. Radovanović, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao, M. Haridasan, P. Hung, N. Care, S. Talukdar, E. Mullen, K. Smith, M. Cottman, and W. Cirne, "Carbon-aware computing for datacenters," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1270–1280, Mar. 2023, doi: 10.1109/TPWRS.2022.3173250.

[10] Z. Cao, X. Zhou, H. Hu, Z. Wang, and Y. Wen, "Toward a systematic survey for carbon neutral data centers," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 895–936, 2022, doi: 10.1109/COMST.2022.3161275.

[11] F. Constante, F. Silva, and A. Pereira, "DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS," *Mendeley Data*, V5, 2019, doi: 10.17632/8gx2fvg2k6.5.

[12] U.S. Environmental Protection Agency, *Emission Factors for Greenhouse Gas Inventories (GHG Emission Factors Hub)*. Washington, DC, USA, Jan. 2025. [Online]. Available: https://www.epa.gov/climateleadership/ghg-emission-factors-hub

[13] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008, doi: 10.1145/1327452.1327492.

[14] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache Spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016, doi: 10.1145/2934664.

[15] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing," in *Proc. NetDB Workshop*, Athens, Greece, 2011, pp. 1–7.

[16] M. Armbrust, T. Das, J. Torres, B. Yavuz, S. Zhu, R. Xin, A. Ghodsi, I. Stoica, and M. Zaharia, "Structured Streaming: A declarative API for real-time applications in Apache Spark," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, Houston, TX, USA, 2018, pp. 601–613, doi: 10.1145/3183713.3190664.

[17] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache Flink: Stream and batch processing in a single engine," *IEEE Data Engineering Bulletin*, vol. 38, no. 4, pp. 28–38, 2015.

[18] N. R. Herbst, S. Kounev, and R. Reussner, "Elasticity in cloud computing: What it is, and what it is not," in *Proc. 10th Int. Conf. Autonomic Computing (ICAC)*, San Jose, CA, USA, 2013, pp. 23–27.

[19] T. Lorido-Botrán, J. Miguel-Alonso, and J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," *Journal of Grid Computing*, vol. 12, no. 4, pp. 559–592, 2014, doi: 10.1007/s10723-014-9314-7.

[20] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007, doi: 10.1109/MC.2007.443.
[21] B. Acun, B. Lee, F. Kazhamiaka, K. Maeng, U. Gupta, M. Chakkaravarthy, D. Brooks, and C.-J. Wu, "Carbon Explorer: A holistic framework for designing carbon aware datacenters," in *Proc. 28th ACM Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, vol. 2, Vancouver, BC, Canada, 2023, pp. 118–132, doi: 10.1145/3575693.3575754.
[22] Y. Ran, H. Hu, X. Zhou, and Y. Wen, "DeepEE: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning," in *Proc. IEEE 39th Int. Conf. Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019, doi: 10.1109/ICDCS.2019.00070.
[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30*, Long Beach, CA, USA, 2017, pp. 3146–3154.
[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[27] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, Long Beach, CA, USA, 2017, pp. 4765–4774.
[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[29] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ, USA: Wiley, 2007.
[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, A. A. Rusu, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.
[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.
[32] J. Gijsbrechts, R. N. Boute, J. A. Van Mieghem, and D. J. Zhang, "Can deep reinforcement learning improve inventory management? Performance on lost sales, dual-sourcing, and multi-echelon problems," *Manufacturing & Service Operations Management*, vol. 24, 2022, doi: 10.1287/msom.2021.1064.
[33] A. Oroojlooyjadid, M. Nazari, L. V. Snyder, and M. Takáč, "A deep Q-network for the beer game: Deep reinforcement learning for inventory optimization," *Manufacturing & Service Operations Management*, vol. 24, no. 1, pp. 285–304, 2022, doi: 10.1287/msom.2020.0939.
[34] Z. Kegenbekov and I. Jackson, "Adaptive supply chain: Demand–supply synchronization using deep reinforcement learning," *Algorithms*, vol. 14, no. 8, Art. no. 240, 2021, doi: 10.3390/a14080240.
[35] R. Tian, M. Lu, H. P. Wang, B. Wang, and Q. X. Tang, "IACPPO: A deep reinforcement learning-based model for warehouse inventory replenishment," *Computers & Industrial Engineering*, vol. 187, Art. no. 109829, 2024, doi: 10.1016/j.cie.2023.109829.
[36] K. Wang, C. Long, D. Ong, J. Zhang, and X.-M. Yuan, "Single-site perishable inventory management under uncertainties: A deep reinforcement learning approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10807–10813, 2023, doi: 10.1109/TKDE.2023.3241087.
[37] J. Zhang, Y. Zhao, W. Xue, and J. Li, "Vehicle routing problem with fuel consumption and carbon emission," *International Journal of Production Economics*, vol. 170, pp. 234–242, 2015, doi: 10.1016/j.ijpe.2015.09.031.
[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[39] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, Nov. 1999, doi: 10.1109/4235.797969.
[40] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004, doi: 10.1287/opre.1030.0065.
[41] Y. Yang, W. Ingwersen, T. Hawkins, M. Srocka, and D. Meyer, "USEEIO: A new and transparent United States environmentally-extended input-output model," *Journal of Cleaner Production*, vol. 158, pp. 308–318, 2017, doi: 10.1016/j.jclepro.2017.04.150.
[42] W. W. Ingwersen, M. Li, B. Young, J. Vendries, and C. Birney, "USEEIO v2.0, the US environmentally-extended input-output model v2.0," *Scientific Data*, vol. 9, Art. no. 194, 2022, doi: 10.1038/s41597-022-01293-7.
[43] W. Ingwersen, "Supply Chain Greenhouse Gas Emission Factors v1.3 by NAICS-6," U.S. Environmental Protection Agency, Washington, DC, USA, 2024. [Online]. Available: https://catalog.data.gov/dataset/supply-chain-greenhouse-gas-emission-factors-v1-3-by-naics-6
[44] J. Heiss, T. Oegel, M. Shakeri, and S. Tai, "Verifiable carbon accounting in supply chains," *IEEE Transactions on Services Computing*, 2023, doi: 10.1109/TSC.2023.3332831.
[45] T. K. Agrawal, V. Kumar, R. Pal, L. Wang, and Y. Chen, "Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry," *Computers & Industrial Engineering*, vol. 154, Art. no. 107130, 2021, doi: 10.1016/j.cie.2021.107130.
[46] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.
[47] Q. Li, Y. Cui, T. Song, and L. Zheng, "Federated multiagent actor–critic learning task offloading in intelligent logistics," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11696–11707, 2023, doi: 10.1109/JIOT.2023.3244557.
[48] J. D. C. Little, "A proof for the queuing formula: L = λW," *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961, doi: 10.1287/opre.9.3.383.
[49] U.S. Environmental Protection Agency, *Emissions & Generation Resource Integrated Database (eGRID) With 2023 Data, Summary Tables, Revision 2*. Washington, DC, USA, Jun. 2025. [Online]. Available: https://www.epa.gov/egrid/summary-data
[50] K. R. Ahmed, M. E. Ansari, M. N. Ahsan, A. Rohan, M. B. Uddin, and M. A. H. Rivin, "Deep learning framework for interpretable supply chain forecasting using SOM ANN and SHAP," *Scientific Reports*, vol. 15, Art. no. 26355, 2025, doi: 10.1038/s41598-025-11510-z.