

Rolling Conformal Prediction for Trigger-Aware Dynamic Rescheduling of Battery-Pack Production under Material-Arrival Uncertainty

Daiki Morita, Ayaka Shimizu, Ren Kobayashi *

Tokyo University of Science, Tokyo, Japan

Abstract: Material-arrival uncertainty in battery-pack assembly is not stationary: inbound performance drifts with carrier capacity, seasonal peaks and supplier disruptions, so a production schedule that was well protected last month may be unprotected today. Data-driven scheduling pipelines typically calibrate the arrival-delay model once, offline, and then either freeze the plan or reschedule on a fixed clock. This paper proposes RCDR, a rolling-conformal, trigger-aware rescheduling framework that keeps the uncertainty model calibrated online and uses the calibration state itself as the disruption signal. Conditional median and upper-quantile arrival models are trained under a strict leakage-controlled chronological protocol; adaptive conformal inference updates the protective buffer at every realised arrival; and an exponentially weighted control chart on the conformal miscoverage indicator, combined with a buffer-shift test and an incremental plan-health check, decides when a four-objective frozen-prefix NSGA-II reoptimisation is worth its cost. On 67,138 held-out real inbound records containing genuine distribution shift, a frozen split-conformal bound drifts between 63% and 98% rolling coverage, whereas the rolling calibration holds 90.0% coverage with a worst half-month block coverage of 89.9% and 12.7% lower pinball loss. Across 189 rolling-horizon runs spanning three real logistics regimes, RCDR matches periodic rescheduling on realised weighted tardiness (-1.2%, $p = 0.750$) while using 32.8% fewer reoptimisations and generating 33.4% less schedule nervousness (both $p < 0.001$), and it improves on a classical deviation trigger by 13.3% in tardiness ($p = 0.005$). Calibration monitoring costs 0.28% of optimiser time.

Keywords: Battery-pack manufacturing, dynamic rescheduling, adaptive conformal inference, distribution shift, multi-objective scheduling, conditional value-at-risk.

1. Introduction

A battery pack is assembled from components that are neither interchangeable nor simultaneously required. Cell lots gate incoming inspection and grading, battery-management-system (BMS) hardware gates electrical integration, and enclosure and cooling parts gate pack closure [1-3]. The production consequence of a late inbound shipment therefore depends on which gate it feeds and on where the affected job sits in the sequence. A one-day slip on an enclosure delivery can be absorbed if the job is scheduled late in the shift, and can idle five stages if it is not.

Two research streams address this situation from opposite sides. Supply-chain analytics predicts arrival dates or late-delivery probabilities, while production scheduling assumes deterministic release dates or protects them with a single global safety buffer. A recent data-to-decision framework for battery-pack manufacturing closes that gap elegantly: it learns leakage-controlled conditional arrival quantiles, converts prediction residuals into stratified scenarios, and feeds them to a four-objective flexible-flow-shop optimiser that minimises expected makespan, expected weighted tardiness, energy and the conditional value-at-risk (CVaR) of material-induced waiting, with an auditable experiment package [4]. That study demonstrates convincingly that conditional upper quantiles dominate a uniform buffer and that risk and timeliness are genuinely conflicting objectives.

It also, deliberately, solves a static problem: the arrival model is calibrated once, the schedule is computed once, and both are then frozen. In a real plant neither assumption survives contact with the inbound stream. Using 96,440 real delivery records from a public marketplace logistics dataset,

we find that the share of shipments arriving later than a frozen 90%-conformal upper bound moves from 2.4% in one half-month block to 30.5% in another within a nine-month window, driven by peak-season congestion and carrier capacity. A frozen calibration is thus wrong in two opposite ways at different times. When it under-covers, the scenario set understates the tail, the CVaR objective is optimised against a fiction, and the line stops. When it over-covers, the plan reserves slack that is never needed, and makespan, tardiness and energy all suffer. Neither failure announces itself as a discrete "disruption event".

Classical rescheduling policies do not observe this. Periodic policies reoptimise on a fixed clock regardless of whether anything has changed, paying computation and, more importantly, nervousness: every revision invalidates material staging, operator instructions and downstream commitments [5-7]. Event-driven policies fire when an individual realisation deviates from plan [8], which by construction happens after the damage has been done and treats a single outlier and a systematic regime change identically. Hybrid and stability-aware variants trade the two off [9-11], but the trigger always looks at the realised schedule, never at the statistical health of the model that produced it.

This paper takes the opposite view. Under conformal prediction the sequence of miscoverage indicators is, by construction, a Bernoulli stream with a known target rate. It is therefore directly monitorable with standard statistical process control, and a shift in it is evidence that the uncertainty model, not merely one shipment, has changed. We build a rescheduling framework, RCDR (Rolling Conformal Drift-triggered Rescheduling), around that observation.

The contributions are as follows. (i) We formulate battery-

pack rescheduling as a rolling-horizon decision process in which the arrival-uncertainty model is re-calibrated online by adaptive conformal inference (ACI), so the scenario set consumed by the optimiser tracks the current inbound regime instead of a historical one. (ii) We propose a composite calibration-state trigger that combines an exponentially weighted moving-average (EWMA) control chart on the conformal miscoverage indicator, a test on the movement of the calibrated protective buffer, and an incremental plan-health check; the whole monitor costs 0.28% of one reoptimisation. (iii) We validate the calibration layer on two independent real logistics datasets, one genuinely non-stationary and one effectively stationary, and show that rolling calibration restores near-exact local coverage in the first without degrading the second. (iv) We evaluate seven rescheduling policies over 189 rolling-horizon runs in three real inbound regimes selected by the behaviour of the frozen conformal bound, with paired significance tests. (v) We release the complete code, raw observations and fixed seeds; every arrival deviation used in the study is an observed value from a public dataset, and the shop-floor layer is transparent, seeded and synthetic by design.

Section II reviews related work, Section III formalises the problem, Section IV describes RCDR, Sections V and VI present the experimental design and results, and Sections VII and VIII discuss implications and limitations.

2. Related Work

2.1. Predictive-Reactive Scheduling and Rescheduling Policies

Dynamic scheduling surveys distinguish completely reactive dispatching, predictive-reactive rescheduling and robust proactive scheduling [5], [12]. Within predictive-reactive schemes, the rescheduling policy answers when to revise, and the rescheduling method answers how [6]. Church and Uzsoy compared periodic and event-driven policies and established the basic trade-off between responsiveness and stability [7]; Sabuncuoglu and Bayiz analysed reactive policies under machine breakdowns and showed that rescheduling frequency interacts strongly with shop load [8]. Rolling-horizon formulations restrict revision to the not-yet-started portion of the plan [13].

Because each revision imposes an organisational cost, stability has become an objective in its own right. Rangsaritratamee et al. optimise efficiency and stability simultaneously [9], and Pfeiffer et al. evaluate rescheduling strategies by simulation with explicit stability metrics [10]. Aytug et al. survey execution under uncertainty and note that most triggers are defined on realised deviations rather than on any model of the uncertainty [11]. Cowling and Johansson argue for using real-time information selectively rather than continuously [14], which is precisely the question a good trigger must answer. Our contribution belongs to this literature but changes the observable: the statistic being monitored is the calibration of the predictive model, not the deviation of the schedule.

2.2. Risk-Aware and Energy-Aware Production Scheduling

Robust optimisation protects decisions against uncertainty sets [15], while CVaR provides a coherent tail-risk measure that is attractive when rare but severe delays dominate the cost [16]. Modos et al. study robust scheduling under energy-

consumption limits [17], and multi-objective energy-aware flexible-job-shop models expose the conflict between completion performance and power draw [18]. Evolutionary multi-objective optimisation, and NSGA-II in particular, remains the standard vehicle for such problems because non-dominated sorting and crowding preserve a diverse Pareto approximation [19]; flexible-job-shop metaheuristics provide the encoding conventions for joint sequencing and machine assignment [20], [21]. These formulations assume the scenario set is given and correct. The present work asks what happens when it is not, and when it stops being correct halfway through the horizon.

2.3. Conformal Prediction under Distribution Shift

Conformal prediction converts any point or quantile predictor into a set predictor with finite-sample validity under exchangeability [22], [23]. Conformalised quantile regression (CQR) applies this to conditional quantile models and inherits their adaptivity to heteroscedasticity [24]. Exchangeability is exactly what a drifting inbound stream violates. Weighted conformal prediction handles known covariate shift [25]; adaptive conformal inference (ACI) instead updates the nominal level online, guaranteeing long-run coverage without any assumption on how the distribution moves [26]. Extensions refine the update rule for time series and for arbitrary shifts [27-29], and theory now characterises conformal validity beyond exchangeability [30]. Calibration in the broader sense is a recognised prerequisite for decision-making with learned models [31].

Machine learning has been coupled to production rescheduling before [32], and tree-based models have been used to predict availability dates under supply disruption [33]. Learning-based dispatching offers fast online decisions but needs large simulator budgets and typically optimises a single objective [34-36]. To our knowledge, the calibration state of a conformal predictor has not previously been used as the rescheduling trigger, nor has online conformal calibration been embedded in a multi-objective production rescheduling loop.

3. Problem Formulation

3.1. Battery-Pack Flexible Flow Shop

Let $J = \{1, \dots, n\}$ be battery-pack jobs and $S = \{1, \dots, 6\}$ ordered assembly stages: (1) incoming inspection and cell grading, (2) grouping and mechanical positioning, (3) interconnection and welding, (4) BMS and electrical integration, (5) enclosure and cooling integration and closure, and (6) end-of-line testing. Each stage offers two alternative machines, k in $\{0, 1\}$: machine 0 is faster and draws more power, machine 1 is slower by 16% and is energy-efficient. Three material classes $M = \{\text{cells, BMS, enclosure}\}$ gate stages 1, 4 and 5 respectively. Job j belongs to family $g(j)$ in $\{0, 1, 2\}$, has due date $d(j)$ and tardiness weight $w(j)$ in $\{1, 2, 3\}$; a family change on a machine incurs a stage-specific setup time and setup energy. This staged abstraction with alternative equipment, family setups and material gates follows the transparent benchmark design of [4], which we adopt so that the manufacturing layer is comparable with the static case.

A plan is a pair (π, X) : a job permutation π applied at every stage and a binary machine-assignment matrix $X = [x(j, s)]$. Given an arrival vector a , the start of operation (j, s) is

$$B(j, s) = \max \{C(j, s-1), M(s, x(j, s)), a(j, m) \mid [s = s(m)]\} \quad (1)$$

And $C(j, s) = B(j, s) + u(j, s) + p(j, s, x(j, s))$, where M is machine availability and u the family-change setup. Material-induced waiting accumulates only the positive difference between material readiness and the otherwise feasible start, so machine congestion is not double-counted as supply risk.

3.2. Arrivals, Residuals and Information Revelation

Each job-material pair (j, m) is bound to one real inbound record with covariates $x(j, m)$ observable before arrival and realised promise deviation $y(j, m)$ in days. The planner does not schedule against the raw promise but against the conditional median forecast: the planned availability is

$$a\text{-hat}(j, m) = t_0(j, m) + q_{0.5}(x(j, m)) \quad (2)$$

Where t_0 is the promised date implied by the nominal plan, so the planning-relevant random variable is the residual $r(j, m) = y(j, m) - q_{0.5}(x(j, m))$. Realised availability is $a(j, m) = \max \{ a\text{-hat}(j, m) + \kappa r(j, m), 0 \}$ with κ a days-to-hours modelling scale. Positive residuals create material waiting; negative residuals are free.

Information is revealed asymmetrically, as in practice: an early shipment is known when it physically arrives, and a late shipment is announced at its planned date. Formally the status of (j, m) becomes known at $\min \{ a\text{-hat}(j, m), a(j, m) \}$. Before that instant the planner has only $x(j, m)$ and the current calibration state; after it, both $a(j, m)$ and the nonconformity score of the record are available as feedback.

3.3. Rolling-Horizon Decision Process

The horizon is divided into decision epochs $T(0) < T(1) < \dots$ spaced by an interval δ . At epoch $T(k)$ the planner holds an incumbent plan, observes all arrivals revealed so far, and may reoptimise. Execution imposes non-anticipativity through a frozen prefix: jobs whose first operation has already started under the incumbent plan keep their positions and machine rows, and the frozen set grows monotonically. Only the unfrozen suffix is re-decided. The realised performance of a policy is obtained by evaluating the final plan, which contains every commitment made along the way, under the fully realised arrival vector.

3.4. Objectives and Stability

For a scenario set Ω the optimiser minimises expected makespan f_1 , expected weighted tardiness f_2 , energy f_3 (deterministic for a fixed plan, since arrival uncertainty changes idle time but not processing duration or machine choice), and the CVaR at level β of total material-induced waiting W :

$$f_4 = \text{CVaR}(\beta)[W] = \text{mean} \{ W(x_i) : W(x_i) \geq Q(\beta)(W) \} \quad (3)$$

Schedule nervousness is not optimised but measured, as the accumulated displacement of unfrozen jobs plus machine reassignments between consecutive plans, following the stability literature [9], [10]. Treating it as an outcome rather than an objective keeps the optimiser comparable across policies and isolates the effect of the trigger, which is what determines nervousness in a predictive-reactive scheme.

4. The RCDR Framework

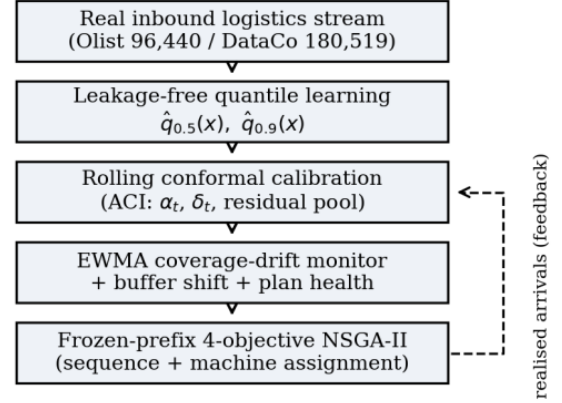


Figure 1. RCDR architecture. Realised arrivals feed back into the rolling calibration and into the trigger, so both the scenario set and the decision to reoptimise adapt to the current inbound regime.

4.1. Leakage-Controlled Conditional Quantile Learning

Two histogram gradient-boosting regressors with pinball loss estimate $q_{0.5}(x)$ and $q_{0.9}(x)$ [37]. Only fields available before the arrival is realised enter the design matrix; every post-arrival field (actual transit time, delivery status, late-delivery labels, carrier hand-over timestamps) is excluded. Splits are strictly chronological with no shuffling, and the stream is ordered by the moment feedback becomes available, i.e. by physical arrival, which is the order in which an online calibrator would actually see it. Categorical variables are ordinal-encoded with an explicit unknown category and numeric variables median-imputed; hyper-parameters are fixed before any evaluation on the stream.

4.2. Rolling Conformal Calibration

Let $e(i) = y(i) - q_{0.9}(x(i))$ be one-sided nonconformity scores. Static split conformal computes a single offset from a calibration block,

$$\delta = \text{Quantile}[\text{ceil}((n+1)(1-\alpha))/n](\{e(i)\}) \quad (4)$$

And freezes it. RCDR replaces this by an online rule. A sliding window of the W most recent observed scores is maintained, and the nominal level itself is updated after each realisation by adaptive conformal inference [26]:

$$\alpha(t+1) = \text{clip}(\alpha(t) + \gamma(\alpha - \text{err}(t))) \quad (5)$$

With $\text{err}(t) = 1[y(t) > q_{0.9}(x(t)) + \delta(t)]$ and $\delta(t)$ the $(1-\alpha(t))$ empirical quantile of the window. When the stream turns adverse the observed miscoverages push $\alpha(t)$ down, the effective quantile level up, and the buffer widens; when the stream calms, $\alpha(t)$ rises and the buffer tightens. The long-run coverage guarantee of ACI requires no assumption on the nature of the shift, which suits inbound logistics, where the cause of a regime change is rarely observable in the purchase-order record. We also report a rolling-window split-conformal variant (window refreshed, α fixed) to separate the contribution of refreshing the calibration data from that of adapting the level.

4.3. Calibration-State Monitoring and the Composite Trigger

The miscoverage indicators $\text{err}(t)$ form a Bernoulli sequence with target mean α , which makes them a natural

control-chart statistic. RCDR maintains

$$z(t) = (1 - \lambda) z(t-1) + \lambda \text{err}(t) \quad (6)$$

With $z(0) = \alpha$ and two-sided limits $\alpha \pm L s(t)$, where $s(t)$ is the exact standard deviation of $z(t)$ under the nominal Bernoulli(α) model. Breaching the upper limit means the plan is under-protected; breaching the lower limit means it is over-protected and capacity is being wasted. Both invalidate the scenario set the incumbent plan was optimised for, so both fire.

Two further terms complete the trigger. The buffer-shift term fires when the calibrated protective buffer, the $(1 - \alpha(t))$ quantile of the current residual pool, has moved by more than τ days since the last reoptimisation; this catches gradual drift that never produces a discrete alarm. The plan-health term evaluates the incumbent plan on the current scenario set and fires when any objective has degraded by more than ϵ relative to its value at the last reoptimisation; this catches the case where the calibration is stable but the particular arrivals revealed so far have hurt this particular plan. Evaluating the incumbent costs one scenario sweep, three orders of magnitude less than a search: measured over all runs, monitoring consumed 0.28% of optimiser time.

4.4. Scenario Construction

At each reoptimisation the scenario set contains $S = 12$ arrival matrices. Revealed gates are fixed at their realised values. Unrevealed gates receive eleven residuals resampled from the current pool plus one conformal stress scenario at $q_{0.9}(x) + \delta(t)$, so the upper tail is always represented explicitly rather than left to sampling luck. For static-calibration policies the pool and δ come from the frozen calibration block; for RCDR they come from the rolling window and the current $\alpha(t)$. This is the only structural difference between the two calibration families, which makes the ablation clean.

4.5. Frozen-Prefix Reoptimisation

The suffix is reoptimised by NSGA-II [19] encoding the permutation of unfrozen jobs and their machine matrix. Order crossover preserves permutation validity, uniform crossover acts on machine bits, binary tournaments use non-domination rank and crowding distance, and the incumbent plan is injected into the initial population so that a revision can never start from scratch. The delivered schedule is the non-dominated solution with minimum root-mean-square normalised distance to the within-front ideal point, applied identically to every policy so that differences reflect calibration and triggering rather than solution selection.

Algorithm 1. RCDR at decision epoch $T(k)$

Step	Operation
1	Ingest arrivals revealed since $T(k-1)$; for each, update the ACI level, the score window and the EWMA chart.
2	Evaluate the incumbent plan on the current scenario set (one sweep).
3	Fire if an EWMA limit was breached, or $ \text{buffer} - \text{buffer_ref} > \tau$, or any objective degraded by more than ϵ .
4	If fired: freeze jobs already started; build 11 residual scenarios + 1 conformal stress scenario.
5	Run NSGA-II on the unfrozen suffix, warm-started from the incumbent; select the compromise solution.
6	Reset the chart and the reference values; otherwise keep executing the incumbent plan.

5. Experimental Design

5.1. Data and Provenance

Two independent public logistics datasets provide every arrival deviation used in this study. The primary source is the Brazilian E-Commerce Public Dataset by Olist [38], from which 96,440 delivered orders with valid timestamps are retained; the target is the promise deviation, delivered date minus estimated delivery date, in days. Predictors are the promised lead time, payment-approval latency, item count, price, freight value, weight, volume, seller count, photo count, purchase month, weekday and hour, freight-to-price ratio, same-state indicator, product category, customer state and seller state. The chronological split is 20,860 training records (to 1 Sept 2017), 8,442 calibration records (to 1 Nov 2017) and a 67,138-record evaluation stream ordered by physical arrival.

The control source is the DataCo Smart Supply Chain dataset [39], 180,519 records, with delay defined as realised minus scheduled shipping days; the file used has SHA-256 994b3c8d24049cc46bf20e8161fedec9a9b95376cc0dc929921d76cf6b9bc0d, identical to the artefact used in [4], so the two studies are directly comparable on this dataset. Splits are 124,604 / 15,455 / 40,460.

5.2. Real Inbound Regimes

Rather than injecting synthetic drift, we select three contiguous windows of the Olist evaluation stream by the behaviour of the frozen split-conformal bound. R1-STRESS (16 Dec 2017 - 16 Jan 2018, 6,132 records) has 30.5% miscoverage against a nominal 10%; the frozen bound is badly under-protective. R2-NOMINAL (16 Jan - 16 Feb 2018, 6,564 records) has 11.5% miscoverage: the frozen bound happens to be right. R3-SLACK (16 Jun - 1 Aug 2018, 9,143 records) has 2.7% miscoverage: the frozen bound is over-protective. The three windows therefore span the two ways a static calibration fails plus the case where it does not, and every residual consumed by the simulation is an observed value in the original chronological order.

5.3. Shop-Floor Benchmark

The manufacturing layer is synthetic, seeded and disclosed in full (Table I); it is not claimed to reproduce measured battery-plant cycle times. Each of the three job sizes is instantiated with three independent seeds, giving nine instances per regime and 27 in total; with seven policies this yields 189 rolling-horizon runs.

Table I. Fixed-Seed Battery-Pack Shop-Floor Design

Element	Specification
Stages / machines	6 ordered stages, 2 alternatives per stage
Material gates	cells: st. 1; BMS: st. 4; enclosure: st. 5
Families / setups	3 families; stage setup 0.08-0.18 h
Base processing	0.65-1.25 h before family, machine and lognormal multipliers
Machines	efficient machine +16% time, 0.62x power
Job sizes / seeds	n in {16, 24, 32}; seeds 101, 203, 307
Epochs	$\delta = 3$ h, at most 14 epochs
Optimiser	NSGA-II, population 30, 20 generations, $S = 12$, $\beta = 0.80$
Uncertainty scale	$\kappa = 0.25$ production hours per logistics day

5.4. Policies Compared

Seven policies form a two-factor design over calibration (frozen split conformal, CP, versus rolling ACI) and trigger (none, periodic, deviation, drift). SB-STATIC plans once against a single global-buffer scenario, the industrial baseline. CP-STATIC plans once against the frozen conformal scenario set; this is the closest analogue of the static framework of [4] inside our rolling-horizon harness. CP-PER reoptimises at every epoch with frozen calibration. CP-DEV reoptimises when a revealed arrival exceeds its predicted upper bound, the classical deviation trigger. CP-DRIFT uses the composite trigger with frozen calibration, and ACI-PER uses rolling calibration with a periodic trigger; these two isolate the individual contributions. RCDR combines both.

5.5. Metrics and Protocol

Realised makespan, realised weighted tardiness, energy

Table II. Calibration of the 90% Arrival Bound on Two Real Streams

Method	Cov.	Worst	Best	MAD	Buf.	Pinb.
Olist stream (67,138 records, real drift)						
Global buffer	0.754	0.550	0.878	0.135	8.11	2.284
Static split CP	0.853	0.768	0.973	0.063	9.94	1.970
Rolling-window CP	0.890	0.869	0.908	0.011	12.25	1.721
ACI (proposed)	0.900	0.899	0.901	0.000	12.68	1.719
DataCo stream (40,460 records, stationary)						
Global buffer	0.922	0.855	0.937	0.025	1.450	0.259
Static split CP	0.961	0.931	0.968	0.059	1.511	0.174
Rolling-window CP	0.961	0.931	0.968	0.059	1.511	0.174
ACI (proposed)	0.900	0.891	0.901	0.001	1.511	0.174

Worst and Best are the extreme half-month block coverages, MAD the mean absolute deviation of block coverage from 0.90, Buf. the mean buffer above the median forecast in days, and Pinb. the pinball loss at level 0.90 (lower is better).

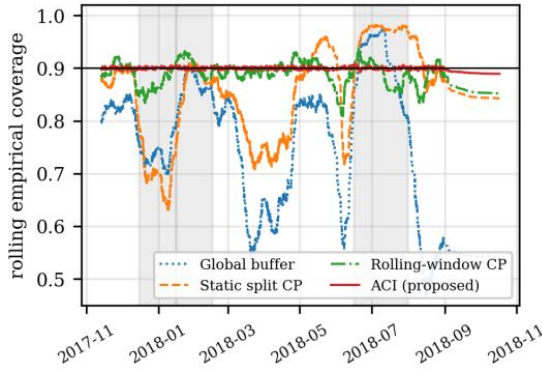


Figure 2. Rolling empirical coverage (window 1,500) of the four bounds on the Olist stream. Shading marks the three regimes used in the scheduling study. The frozen bounds swing between 63% and 98%; the rolling calibration tracks the nominal 90% line throughout.

Table II reports the calibration layer. On the drifting Olist stream the conditional model is worth having: median absolute error falls from 7.396 to 5.607 days (-24.2%) against a constant trained median. But conditioning alone does not deliver calibration. The frozen split-conformal bound attains 85.3% marginal coverage and, more damagingly, block coverages between 76.8% and 97.3%; a global buffer is worse still at 75.4% marginal and 55.0% worst-block. Refreshing the calibration window recovers marginal coverage (89.0%) and cuts the block deviation by a factor of six. Adding the adaptive level closes the gap almost exactly: ACI attains 90.0%

and realised total material waiting are computed from the final plan under the true arrival vector. The delivered plan is additionally stress-tested on 60 residual draws from a disjoint, later part of the same regime, giving an ex-post CVaR at level 0.90 of material waiting; optimisation and this evaluation therefore never share residual observations. Effort is measured by the number of reoptimisations and optimiser CPU time, and nervousness by the instability index of Section III-D. A normalised four-objective compromise score, min-max normalised within each regime and job size and averaged, supports compact comparison. Significance is assessed with paired one-sided-free Wilcoxon signed-rank tests over matched instances ($n = 9$ per regime, $n = 27$ pooled).

6. Results

6.1. Calibration under Real Distribution Shift

marginal coverage with block coverages confined to [0.899, 0.901], a mean absolute block deviation of 0.0004, and the lowest pinball loss (1.719, i.e. 12.7% below static CP and 24.7% below the global buffer). Sharper and better calibrated is unusual; it happens here because the frozen bound is simultaneously too tight in some months and too loose in others, so a bound that tracks the regime wins on both counts.

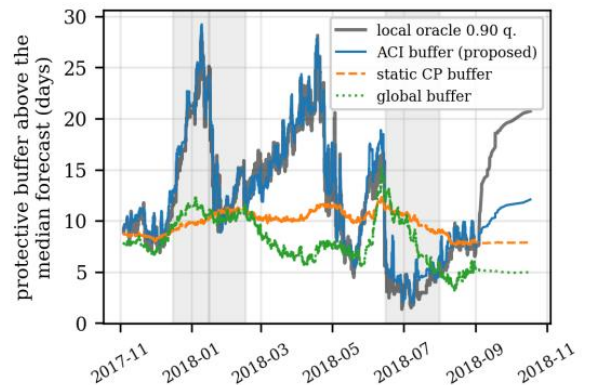


Figure 3. Protective buffer above the median forecast, against the local oracle 0.90 residual quantile. The ACI buffer follows the oracle almost exactly (correlation 0.986) and moves between 3 and 28 days; the frozen conformal buffer barely moves (correlation 0.345).

Figures 2 and 3 make the mechanism visible. The frozen bound is not merely biased, it is alternately over- and under-protective, and the transitions are fast. The ACI buffer, in contrast, tracks the local oracle 0.90 residual quantile with correlation 0.986 against 0.345 for the frozen bound, expanding to 28 days during the December-January crunch and contracting below 4 days in the July calm. This is the

quantity a planner would actually reserve, so its accuracy translates directly into either protected or wasted capacity.

The DataCo panel is the negative control and answers the natural objection that an adaptive method might simply be adding slack. This stream is stationary and, notably, our pipeline reproduces the published static result on it: the median absolute error is 0.984 days against 0.983 reported in [4], and the split-conformal correction is again exactly zero because the raw discrete-target upper quantile is already conservative. The frozen bound over-covers at 96.1%, and ACI tightens it to exactly 90.0% at essentially unchanged pinball loss. Rolling calibration therefore does not act as a blanket safety margin; it moves the bound in whichever direction the data require, and where the data require nothing it converges to the nominal level.

6.2. Sensitivity of the Rolling Calibration

Table III shows that the calibration layer is not delicately tuned. Across step sizes from 0.005 to 0.05 and windows from 500 to 4,000 records, marginal coverage stays within [0.898, 0.900] and the worst block coverage never falls below 0.896.

Only an aggressive step of 0.10 begins to over-react, widening the buffer by 20% and losing a little coverage precision, which is the expected behaviour of a first-order online update. The full 15-cell grid is in the released artefacts. In the scheduling experiments the plant-side loop uses a shorter window (400) and a larger step (0.08) because a single plant observes far fewer arrivals per horizon than the global stream.

Table III. Sensitivity of ACI to Step Size and Window

gamma	W	Cov.	Worst	MAD	Buf.	Pinb.
0.005	1500	0.900	0.896	0.0017	12.43	1.699
0.010	1500	0.900	0.898	0.0011	12.51	1.703
0.020	500	0.900	0.898	0.0007	12.66	1.722
0.020	1500	0.900	0.898	0.0008	12.68	1.719
0.020	4000	0.900	0.896	0.0012	12.73	1.727
0.050	1500	0.898	0.896	0.0015	13.67	1.821
0.100	1500	0.892	0.887	0.0074	15.19	2.001

6.3. Dynamic Rescheduling Performance

Table IV. Rolling-Horizon Results, Means over Nine Instances per Regime

Policy	Cmax	WT	Energy	Wait	R	I
R1-STRESS (frozen-bound miscoverage 0.305)						
SB-STATIC	45.39	617.3	1531	33.80	0.00	0.0
CP-STATIC	45.94	620.7	1519	35.05	0.00	0.0
CP-DEV	42.32	446.0	1432	33.05	9.00	26.1
CP-PER	41.98	444.4	1429	37.06	9.89	29.6
CP-DRIFT	42.86	485.4	1440	34.96	6.89	22.3
ACI-PER	42.41	440.4	1420	33.98	10.11	26.0
RCDR	42.51	430.0	1443	31.83	6.11	16.9
R2-NOMINAL (miscoverage 0.115)						
SB-STATIC	42.09	508.5	1508	38.41	0.00	0.0
CP-STATIC	43.44	574.6	1519	33.48	0.00	0.0
CP-DEV	41.35	475.4	1434	30.91	5.44	18.4
CP-PER	41.01	430.5	1443	32.72	9.67	27.2
CP-DRIFT	42.65	483.5	1449	32.34	2.89	9.5
ACI-PER	41.03	410.2	1439	32.80	10.00	26.0
RCDR	41.33	419.2	1437	32.57	8.11	24.3
R3-SLACK (miscoverage 0.027)						
SB-STATIC	41.78	484.0	1503	32.88	0.00	0.0
CP-STATIC	42.22	540.0	1518	29.32	0.00	0.0
CP-DEV	42.03	501.6	1453	28.56	1.89	6.9
CP-PER	40.02	374.3	1439	29.33	9.56	28.7
CP-DRIFT	40.57	465.6	1446	28.45	2.22	8.1
ACI-PER	40.20	370.8	1448	30.98	9.56	30.2
RCDR	39.54	385.1	1454	27.89	5.33	15.6

Cmax realised makespan (h), WT realised weighted tardiness (h), Energy kWh, Wait realised material waiting (h), R reoptimisations per horizon, I instability index.

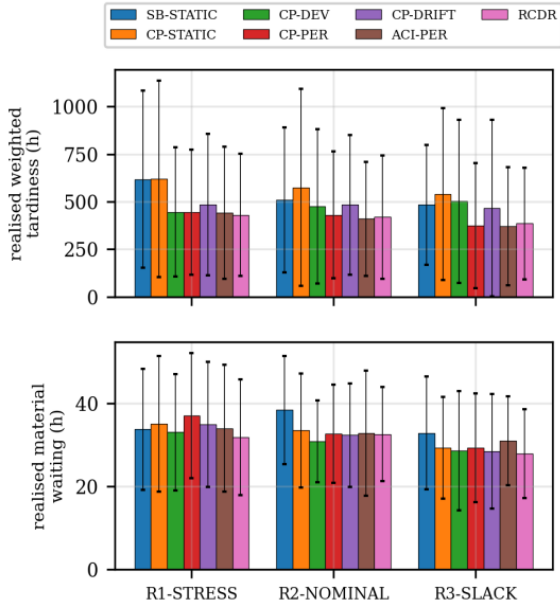


Figure 4. Realised weighted tardiness and realised material waiting by policy and regime; error bars are standard deviations over the nine instances of each regime.

Table IV and Fig. 4 give the operational picture. Rescheduling of any kind is worth far more than the choice of static plan: the two frozen policies lose 23% to 29% of weighted tardiness relative to every adaptive policy, in all three regimes. Among adaptive policies the ranking depends on the regime in an interpretable way. In R1-STRESS, where the frozen bound under-covers, RCDR delivers the lowest realised tardiness (430.0 h) and the lowest realised material waiting (31.83 h) of all seven policies while reoptimising 6.11 times against 9.89 for the periodic policy. In R2-NOMINAL, where the frozen calibration is already correct, the advantage of adapting it shrinks as expected and RCDR is statistically indistinguishable from the periodic policies. In R3-SLACK it recovers the leading position on makespan (39.54 h) and waiting (27.89 h) with 5.33 reoptimisations against 9.56.

Energy varies by less than 2% across adaptive policies, which is consistent with the structure of the problem: energy is fixed by the machine assignment, and all policies face the same speed-versus-power trade-off. The ex-post CVaR of the delivered plan likewise differs little between the adaptive policies (45.0 to 46.2 h pooled, no significant pairwise difference), and we report this plainly: the benefit of rolling calibration in our setting shows up in realised timeliness, realised waiting and rescheduling effort rather than in the stress-tested tail of the final plan, because by the end of the horizon most gates are already revealed and the remaining uncertainty is small for every policy.

6.4. Effort, Stability and Significance

Table V. RCDR against Each Baseline, Pooled over 27 Instances

Baseline	dWT%	p	dR%	p	dI%	p
SB-STATIC	-23.3	0.000	n/a	-	n/a	-
CP-STATIC	-28.9	0.000	n/a	-	n/a	-
CP-DEV	-13.3	0.005	+19.7	0.094	+10.7	0.301
CP-PER	-1.2	0.750	-32.8	0.000	-33.4	0.000
CP-DRIFT	-14.0	0.001	+63.0	0.003	+42.8	0.032
ACI-PER	+1.1	0.645	-34.1	0.000	-30.8	0.000

Relative differences of RCDR minus baseline for weighted tardiness (dWT), reoptimisation count (dR) and instability (dI); negative favours RCDR on dWT, and lower dR and dI mean less effort and less nervousness. Paired Wilcoxon signed-rank p-values; n/a marks baselines that never reschedule.

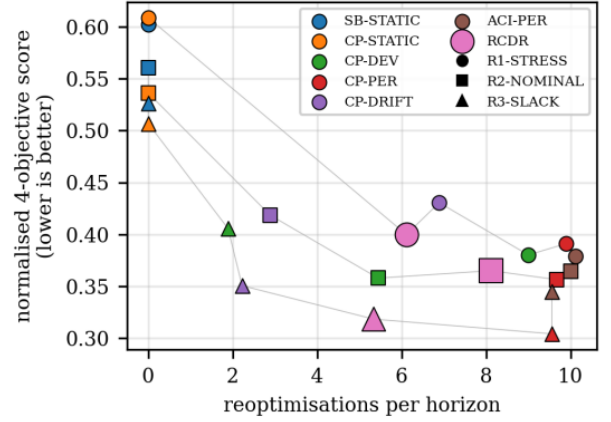


Figure 5. Effort-benefit plane. RCDR (large markers) sits at the knee: it reaches the score of the periodic policies with roughly one third fewer reoptimisations, and clearly dominates the sparser triggers.

Table V quantifies the central claim. Against the two frozen plans, RCDR reduces weighted tardiness by 23.3% and 28.9% (both $p < 0.001$) and the composite score by about 35%. Against the periodic policies it is statistically indistinguishable on tardiness (-1.2%, $p = 0.750$ versus CP-PER; +1.1%, $p = 0.645$ versus ACI-PER) while using 32.8% and 34.1% fewer reoptimisations and producing 33.4% and 30.8% less nervousness, all four differences significant at $p < 0.001$. In other words, monitoring the calibration state buys back a third of the rescheduling effort and a third of the schedule churn at no measurable cost in performance.

Against the classical deviation trigger the comparison is even clearer: RCDR improves tardiness by 13.3% ($p = 0.005$) while the extra effort it spends is not statistically distinguishable (+19.7%, $p = 0.094$). The reason is visible in the regime breakdown. The deviation trigger is driven by individual breaches, so it fires nine times in R1-STRESS, where breaches are frequent, and only 1.9 times in R3-SLACK, where the frozen bound almost never breaches even though it has become far too conservative and is quietly costing 23% in tardiness there. A trigger that watches only for failures is blind to over-protection; a trigger that watches the calibration sees both.

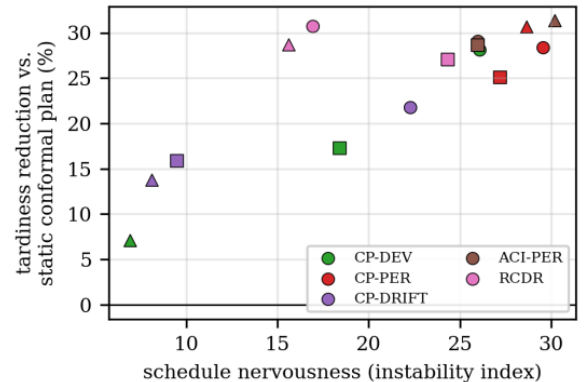


Figure 6. Nervousness against the tardiness improvement over the static conformal plan. Policies towards the upper left convert less schedule churn into more benefit; RCDR occupies that region in all three regimes.

6.5. Ablation

The two components contribute differently. Replacing rolling calibration by the frozen one while keeping the composite trigger (CP-DRIFT) costs 14.0% in tardiness ($p = 0.001$); the trigger still saves effort, but it now protects a scenario set that is itself wrong, and in R3-SLACK it converges to a very sparse schedule of 2.2 revisions that leaves most of the available improvement on the table. Conversely, keeping rolling calibration but firing periodically (ACI-PER) recovers the performance, and indeed is marginally the best single policy on tardiness in R2 and R3, but pays 34.1% more reoptimisations and 30.8% more nervousness. Only the combination sits at the knee of the effort-benefit curve in Fig. 5, and it is the only policy that is simultaneously in the top group on realised waiting in all three regimes.

7. Discussion

7.1. Managerial Implications

Three practical conclusions follow. First, the value of a predictive layer in scheduling lies as much in its calibration as in its accuracy. Our conditional model improves median error by 24% over a constant, but the difference between a frozen and a rolling calibration of the same model changes realised tardiness by up to 14%. A plant that deploys a delay model and never re-examines its coverage is running an uncontrolled process.

Second, the coverage chart is a cheap and interpretable early-warning device. It is computed from data the plant already has, it has a natural target, it is symmetric in the two failure modes, and at 0.28% of the cost of a reoptimisation it can run continuously. It also gives planners a defensible reason to not reschedule, which is organisationally valuable: a periodic policy provides no argument against churn.

Third, over-protection deserves the same attention as under-protection. The R3-SLACK regime is the one in which a conventional deviation trigger performs worst, precisely because nothing appears to be going wrong. A buffer inherited from a bad season silently consumes capacity, and only a two-sided calibration test detects it.

7.2. Validity Boundaries

Four limitations bound these conclusions. First, the inbound datasets are general logistics streams, not battery-component inbound logistics; they support the methodological claim about calibration and triggering, not a transfer of effect sizes to a specific plant. Second, the shop-floor layer, including processing times, power, setups, due dates and families, is synthetic and seeded, and energy is an internally consistent processing-power proxy that excludes HVAC, standby and demand charges. Third, delays are recorded in whole or fractional days and rescaled to production hours by a fixed constant; event-level timestamps would allow a direct calibration. Fourth, the evaluation protocol freezes decisions rather than replaying a full discrete-event execution, and the base learner is not retrained during the horizon, so the reported gains isolate the calibration and triggering layers rather than the combined effect of retraining and recalibration.

Natural extensions are to replace each synthetic layer independently: supplier advance-shipping notices for the arrival model, manufacturing-execution-system logs for processing and setup distributions, meter data for energy, and

contractual penalties for objective weighting. Coupling the coverage chart to a model-retraining decision, so that persistent drift triggers relearning rather than only recalibration, is a further step, as is extending the trigger to multivariate calibration statistics when several material classes drift independently.

8. Conclusion

We proposed RCDR, a rolling-conformal and trigger-aware framework for rescheduling battery-pack production under material-arrival uncertainty. Its two ideas are that the arrival-uncertainty model should be recalibrated online rather than fitted once, and that the calibration state of that model, rather than the realised deviations of the schedule, is the right signal for deciding when to reschedule. On 67,138 real held-out inbound records containing genuine distribution shift, rolling calibration held 90.0% coverage with a worst half-month block coverage of 89.9%, against 85.3% and 76.8% for a frozen split-conformal bound, while also reducing pinball loss by 12.7%; on a stationary control stream it tightened an over-conservative bound to the nominal level instead of adding slack. Across 189 rolling-horizon runs in three real inbound regimes, RCDR matched periodic rescheduling on realised weighted tardiness while using a third fewer reoptimisations and generating a third less schedule nervousness, and improved on a classical deviation trigger by 13.3% in tardiness. The claim is not that more rescheduling is better, but that rescheduling should be governed by a statistic that knows when the uncertainty model has stopped being true.

References

- [1] Koelmel, A., Sauer, A., & Lanza, G. (n.d.). Quality-oriented production.
- [2] Zhang, H. (2024). Graph-aware multi-task learning for early prediction of timing violations and routing congestion in digital IC physical design. *World Journal of Information Technology*, 2(1), 13–20.
- [3] Wang, J., Tan, Y., Jiang, B., Wu, B., & Liu, W. (2025). Dynamic marketing uplift modeling: A symmetry-preserving framework integrating causal forests with deep reinforcement learning for personalized intervention strategies. *Symmetry*, 17(4), 610.
- [4] Zhao, X., Liu, J., Wang, Y., & Wang, J. (2026). CryptoMamba-SSM: Linear complexity state space models for cryptocurrency volatility prediction. *IEEE Open Journal of the Computer Society*, 7, 226–243.
- [5] Wang, J., Liu, J., Zheng, W., & Ge, Y. (2025). Temporal heterogeneous graph contrastive learning for fraud detection in credit card transactions. *IEEE Access*.
- [6] Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. *IEEE Access*, 12, 116546–116557.
- [7] Zhang, H. (2025). Reinforcement learning approaches for layout optimization in electronic design automation with electromagnetic compatibility constraints. *Frontiers in Robotics and Automation*, 2(2), 77–93.
- [8] Han, X., Yang, Y., Chen, J., Wang, M., & Zhou, M. (2025). Symmetry-aware credit risk modeling: A deep learning framework exploiting financial data balance and invariance. *Symmetry*, 17(3), 341.
- [9] Chen, J., Cui, Y., Zhang, X., Yang, J., & Zhou, M. (2024). Temporal convolutional network for carbon tax projection: A data-driven approach. *Applied Sciences*, 14(20), 9213.

- [10] Zhang, C., & Wang, Y. (2024). A data-driven multi-objective production scheduling framework for battery pack manufacturing under material arrival uncertainty. *AI and Data Science Journal*, 1(1), 86–93.
- [11] Chen, J., Wang, M., & Sun, T. (2025). Intelligent tax systems and the role of natural language processing in regulatory interpretation. *American Journal of Machine Learning*, 6(4), 74–94.
- [12] Chen, Z., Liu, J., & Chen, J. (2025). Machine learning methods for financial forecasting in enterprise planning: Transitioning from rule-based models to predictive analytics. *Frontiers in Artificial Intelligence Research*, 2(3), 541–564.
- [13] Chen, J., Liu, J., Liang, Y., & Zhou, M. (2026). KE-MLLM: A knowledge-enhanced multi-sensor learning framework for explainable fake review detection. *Applied Sciences*, 16(6), 2909.
- [14] Yue, X., Yang, J., & Liu, W. (2026). FraudDebate-Agent: A multi-agent LLM framework with an evidence-based debate mechanism for financial statement fraud detection. *Mathematics*, 14(15), 2695.
- [15] Wang, Y., & Zhang, C. (2023). AERIS: Adaptive executor-scoped resource management for privacy-preserving data processing in Apache Spark. *Eurasia Journal of Science and Technology*, 5(2), 25–33.
- [16] Yue, X., & Zhu, R. (2024). An explainable hybrid machine learning framework for cash-flow-conditioned multi-horizon liquidity risk early warning: An SME-oriented benchmark study. *World Journal of Management Science*, 2(1), 63–72.
- [17] Zou, J., Qin, Y., & Rhee, M. (2024). Beyond model scale: A task-aware reliability benchmark for large language models in clinical decision support. *Innovation and Technology Studies*, 1(1), 39–47.
- [18] Zhang, S., & Qiu, L. (2023). An interpretable, uncertainty-aware framework for bridge deterioration prediction and risk-based maintenance prioritization. *Academic Journal of Architecture and Civil Engineering*, 1(2), 21–28.
- [19] Jin, J., Xing, S., Ji, E., & Liu, W. (2025). Xgate: Explainable reinforcement learning for transparent and trustworthy API traffic management in IoT sensor networks. *Sensors*, 25(7), 2183.
- [20] Xing, S., Wang, Y., & Liu, W. (2025). Multi-dimensional anomaly detection and fault localization in microservice architectures: A dual-channel deep learning approach with causal inference for intelligent sensing. *Sensors*, 25(11), 3396.
- [21] Xing, S., & Wang, Y. (2025). Proactive data placement in heterogeneous storage systems via predictive multi-objective reinforcement learning. *IEEE Access*, 13, 117986–117998.
- [22] Ji, E., Wang, Y., Xing, S., & Jin, J. (2025). Hierarchical reinforcement learning for energy-efficient API traffic optimization in large-scale advertising systems. *IEEE Access*.
- [23] Xing, S., & Wang, Y. (2025). Cross-modal attention networks for multi-modal anomaly detection in system software. *IEEE Open Journal of the Computer Society*.
- [24] Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. *Sensors*, 24(24), 7949.
- [25] Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. *Applied Sciences*, 15(2), 951.
- [26] Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. *IEEE Access*, 12, 195974–195985.
- [27] Hu, X., Zhao, X., Wang, J., & Yang, Y. (2025). Information-theoretic multi-scale geometric pre-training for enhanced molecular property prediction. *PLOS One*, 20(10), e0332640, <https://doi.org/10.1371/journal.pone.0332640>.
- [28] Wang, M., Zhang, X., Yang, Y., & Wang, J. (2025). Explainable machine learning in risk management: Balancing accuracy and interpretability. *Journal of Financial Risk Management*, 14(3), 185–198.
- [29] Yang, J., Li, P., Cui, Y., Han, X., & Zhou, M. (2025). Multi-sensor temporal fusion transformer for stock performance prediction: An adaptive Sharpe ratio approach. *Sensors*, 25(3), 976.
- [30] Zhao, X., Sun, T., Ren, S., Yang, J., & Liu, Y. (2025). RAG-based AI agents for enterprise software development: Implementation patterns and production deployment. *Frontiers in Artificial Intelligence Research*, 2(3), 501–520.
- [31] Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. *Asian Business Research Journal*, 10(11), 1–13.
- [32] Bian, M., Wang, Y., & Lin, Y. (2024). Explainable supplier risk prediction for critical telecommunications infrastructure procurement: An empirical XGBoost–SHAP framework. *Social Science and Management*, 1(1), 64–77.
- [33] Rhee, M., Zou, J., Mo, T., Teng, D., & Yang, J. S. (2026). Enhancing web search agents with self-play contrastive fine-tuning. *IEEE Access*.
- [34] Wang, B., Zhao, W., & Wang, Z. (2024). DAS-GNN: A scalable disagreement-aware graph neural framework for anomaly detection in large-scale networks. *AI and Data Science Journal*, 1(1), 67–75.
- [35] Wang, Z., & Shen, Z. (2024). Flex-Talent: An explainable and fair machine learning framework for high-stakes leadership-pipeline evaluation. *World Journal of Management Science*, 2(1), 73–82.
- [36] Guo, Z., Chen, T., & Shang, W. (2022). Class-conditional intermediate-domain adversarial adaptation for cross-device acoustic scene classification. *Journal of Computer Science and Electrical Engineering*, 4(1), 26–35.
- [37] Wang, Y., Bian, M., & Lin, Y. (2022). An explainable temporal graph neural network for disruption prediction and risk propagation in multi-tier supply chains. *Journal of Computer Science and Electrical Engineering*, 4(1), 17–25.
- [38] Zhao, W., & Wang, B. (2022). CAPA: A prediction-driven autoscaling framework for SLO-aware cloud-native machine learning inference. *Journal of Computer Science and Electrical Engineering*, 4(1), 8–16.
- [39] Jiao, Y., & Liang, Y. (2024). An integrated process-mining and explainable machine-learning framework for bottleneck and internal-control risk detection in the multi-entity financial close. *Social Science and Management*, 1(3), 96–107.