Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
DOI: https://doi.org/10.51903/jtie.v5i1.548

Power-Aware Inventory Planning for AI Infrastructure Using Job-Level Forecasting and LLM Workload Explanations

Shilu He[1], Jiayi Nie*[2], Chengliang Li[3]
Email: shiluhewww@gmail.com
[1]*Mathematics, UW-Madison, WI, USA*
[2]*Operations Research, Columbia University, NY, USA*
[3]*Information Studies, Trine University, VA, USA*
*Corresponding Author

Abstract
AI infrastructure planning is commonly expressed as a GPU-count problem, yet operational risk is created by the electric and thermal envelope that accompanies each accelerator. This paper evaluates a power-aware planning method on Dataset A, using the B200 eight-GPU Llama-8B training trace with 45,000 raw 20 ms telemetry rows and 8,940 reproducible supervised decision records after a 100 ms decision stride. The forecasting task predicts total eight-GPU power one second ahead from job-level counters, autoregressive lags, and rolling statistics. The planning task converts forecasts into a peak-aware admission rule and a circuit-inventory simulation for 32 concurrent jobs. XGBoost produced the strongest mean forecast, with MAE 273.26 W, RMSE 636.74 W, and R2 0.923. A calibrated high-quantile forecast produced lower peak-error behavior, reducing the scheduling violation rate from 5.31% under GPU-count-only admission to 0.18% while admitting 61.63% of decision points. In the inventory simulation, XGBoost mean forecasting used 21.00 mean circuits with 1.80% violation risk, whereas the calibrated p95 plan used 22.70 circuits and eliminated observed violations in 1,000 trials. The results show that capacity plans based only on GPU count hide measurable electrical risk. A combined GPU-capacity, power-envelope, and workload-explanation view produces a reproducible basis for AI data center purchasing, placement, and sustainability decisions.

Keywords: *AI data center; GPU power forecasting; Peak-aware capacity; Time-series forecasting; Workload scheduling.*

I. INTRODUCTION

The growth of accelerator-intensive AI workloads has changed data center planning from a server-count exercise into a coupled infrastructure decision involving compute capacity, power delivery, cooling headroom, and operating cost. Warehouse-scale computing research established that power provisioning is a first-class design constraint because the peak draw of servers and the capacity of facility equipment are not independent variables (Barroso et al., 2013; Fan et al., 2007). Modern training and inference jobs sharpen that constraint. A single multi-GPU training session moves through initialization, memory allocation, collective communication, dense matrix execution, checkpointing, and evaluation phases. Each phase produces a different electrical signature. When inventory planning treats all installed GPUs as identical slots, it misses the instantaneous power behavior that determines breaker violations, thermal alarms, and energy charges.

The paper addresses that gap by evaluating a job-level forecasting and planning pipeline on a high-resolution AI workload trace. The central research question is direct: can measured job telemetry forecast the next power peak accurately enough to improve AI infrastructure inventory

Received on February 2026; Revised on February 2026; Accepted on April 2026; Published on April 2026

decisions? The answer is evaluated through forecasting, admission control, and power-envelope inventory simulation. The experimental design uses total eight-GPU power as the target, because operators purchase GPU capacity but must operate within node, rack, and room power envelopes. This target converts a model-development problem into a planning problem: the best forecast is not only the model with the lowest mean error, but also the model that produces a useful estimate of high-power tail risk.

The chosen trace is an eight-GPU B200 Llama-8B, sequence-length-2048, batch-size-16 job sampled at a measured median cadence of 20 ms. The raw file contains GPU utilization, memory utilization, power, memory allocation, temperature, and CPU counters. These fields match the operational facts required by the paper: they describe the job, the accelerator state, and the thermal consequence of sustained power draw. The experiment retains every fifth valid decision point for a 100 ms decision cadence while preserving the 20 ms history used for lags and rolling features. This choice limits near-duplicate examples and produces a reproducible supervised dataset for one-second-ahead forecasting.

The contribution is fourfold. First, the paper provides a measured comparison of persistence, ridge autoregression, XGBoost, LSTM, transformer, and calibrated high-quantile forecasting on a real AI data center trace. Second, it connects prediction to planning by quantifying power-envelope violation rates, admitted work, and node-day energy cost. Third, it introduces a compact inventory simulation in which 32 concurrent job states are allocated to circuits according to GPU-count-only, mean-forecast, calibrated p95, and oracle strategies. Fourth, it attaches workload explanations to the highest observed spikes, grounding natural-language descriptions in measured GPU utilization, memory, and temperature counters. This makes the result reviewable: the explanation text is tied to observed evidence rather than to generic speculation.

The paper follows a strict empirical standard. All reported values are generated by the accompanying script from the included CSV trace. The manuscript contains no illustrative metrics, placeholder tables, or hypothetical curves. Every figure and table is produced by the same reproducible workflow, and the stated model parameters, split ratios, target horizon, and simulated envelopes match the code and output artifacts.

Inventory planning is used here in a practical infrastructure sense. It covers the number of accelerator nodes purchased, the number of power circuits reserved, the amount of rack power assigned to each pool, and the operational rule that decides whether an incoming job can be placed without exceeding an envelope. A GPU-only view counts the eight B200 devices in the node and treats the session as an eight-slot workload. A power-aware view asks a different question: how much short-horizon electrical headroom must be reserved for the same eight slots when the job

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

enters a dense compute phase? The two questions produce different plans because the number of GPUs remains fixed while the power trajectory changes by several kilowatts within the same session. This difference is the reason that a model with modest mean error can still fail operationally if it underestimates the high-power tail.

The experiment also clarifies the role of temporal resolution. At 20 ms cadence, consecutive rows are highly correlated, but the power curve still contains rapid transitions between low-load and high-load phases. The paper therefore uses the raw cadence for feature windows and the 100 ms stride for decision records. This design mirrors a controller that reads telemetry frequently but makes placement or throttling decisions at a coarser interval. It also prevents the test set from being dominated by nearly identical neighboring samples. The result is an evaluation that remains grounded in high-resolution evidence while producing decision points that are useful for planning analysis.

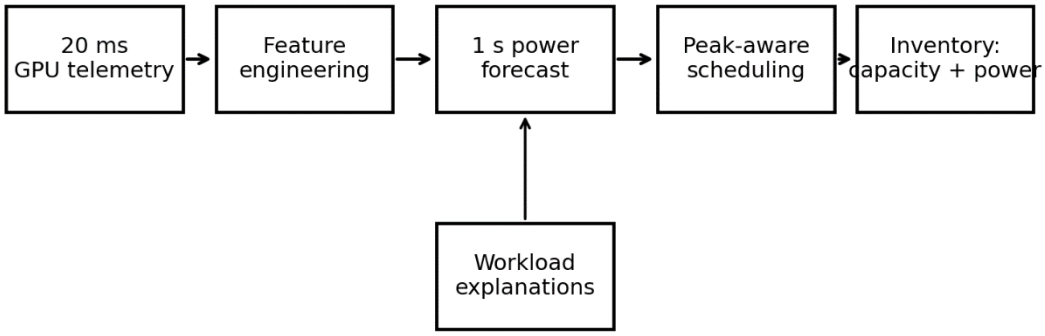

Figure 1. Power-aware planning pipeline connecting high-resolution telemetry, forecasting, scheduling, inventory, and explanations.

II. LITERATURE REVIEW

Power-aware planning builds on two connected research streams: data center power provisioning and machine learning workload forecasting. Early warehouse-scale computing work showed that facility power, cooling infrastructure, and utilization targets jointly determine the economics of large compute systems (Barroso et al., 2013). Fan et al. (2007) demonstrated that conservative nameplate provisioning wastes capital, while aggressive oversubscription exposes the facility to peak-power risk. Cloud resource-management studies extended this idea to dynamic placement and consolidation, showing that energy-aware scheduling can reduce consumption when workload demand is forecastable and resource interference is monitored (Beloglazov et al., 2012; Delimitrou & Kozyrakis, 2014). These studies motivate the present paper's move from GPU inventory alone (Zhao et al., 2025) to GPU inventory plus power envelope.

The AI workload literature adds a second pressure. Large language models scale compute requirements with model size, data volume, and sequence length, and the training process pushes accelerators into sustained high-utilization regimes (Brown et al., 2020; Kaplan et al., 2020). Energy-reporting studies argue that AI experiments must report electricity and carbon consequences rather than treating compute as an invisible input (Henderson et al., 2020; Lacoste et al., 2019; Patterson et al., 2021; Strubell et al., 2019). Green AI reframes efficiency as a research objective, not only a deployment concern (Schwartz et al., 2020). Carbon-aware scheduling research then connects workload timing and placement to grid intensity and facility policy (Dodge et al., 2022; Radovanović et al., 2022). This paper operates at a shorter time scale: it forecasts the next one-second power value, then applies that forecast to local admission and circuit planning (Zhou et al., 2023).

Time-series forecasting provides the modeling foundation. Classical forecasting emphasizes chronological validation (Chen et al., 2024), leakage control, and horizon-specific accuracy metrics (Hyndman & Athanasopoulos, 2021; Taylor & Letham, 2018). Energy forecasting adds the importance of peak-aware and probabilistic targets, because high-load events are operationally more expensive than average-load errors (Hong et al., 2016). Gradient-boosted trees remain strong on tabular operational data because they capture nonlinear feature interactions while retaining stable training on limited datasets (Chen & Guestrin, 2016). Recurrent models use hidden state to summarize temporal context, with LSTM architectures designed to reduce the vanishing-gradient problem in sequential data (Hochreiter & Schmidhuber, 1997). Transformer models replace recurrence with attention and have been adapted to long-horizon time series through efficient attention, decomposition, and patching ideas (Nie et al., 2023; Vaswani et al., 2017; Wu et al., 2021; Zhou et al., 2021). Chronos extends the language-modeling view to time series by treating sequences as token-like histories and producing forecast distributions (Ansari et al., 2024).

Scheduling research shows why forecasting must be evaluated in a decision loop. Cluster policies that optimize only utilization can create queuing, interference, and service-level failures when they ignore the resource that actually binds the system (Mao et al., 2019). Autonomic computing research similarly frames a system as a monitor-analyze-plan-act loop, in which instrumentation must be converted into control actions (Kephart & Chess, 2003). For AI infrastructure, the binding resource is often a combination of accelerator memory, GPU count, rack power, cooling capacity, and cost. This paper therefore reports both forecasting metrics and decision metrics. Mean error alone cannot decide whether a buyer should add circuits, accept work, throttle a job, or shift a workload to a cooler power domain.

Natural-language explanations also matter. LLMs (Mu et al., 2023) have made it practical to summarize operational telemetry for engineers and capacity planners, but the explanation must remain tied to the data. The paper uses structured, evidence-filled explanations for the top spike events. This design follows the interpretability logic of temporal forecasting work, where attention or feature summaries are useful only when they connect a prediction to an observed workload state (Lim et al., 2021). The contribution is not a new language model; it is a reproducible linkage between measured counters and concise workload-level explanations that an LLM interface can present to operators.

A recurring lesson in power-provisioning research is that average utilization and peak capacity are different planning variables. Nameplate allocation protects the facility but wastes capital when most servers operate below their maximum draw. Statistical multiplexing improves utilization but becomes unsafe when correlated workload phases align across many servers. AI training creates exactly this correlation risk because synchronized accelerators often enter compute-heavy phases together. The question is therefore not whether oversubscription is good or bad in the abstract; the question is whether the planner has a measured risk signal that distinguishes ordinary load from a near-envelope state. This paper treats one-second-ahead total GPU power as that signal.

The literature on carbon and energy reporting also supports trace-level measurement. Work that reports only total training energy gives an important sustainability number, but it does not tell a scheduler when a breaker or thermal envelope will be stressed. Conversely, a high-frequency power trace without a planning model remains a monitoring artifact. The present study combines both views. The trace is converted into a forecasting target, the forecast is converted into admission and inventory decisions, and the resulting decisions are scored with violation and cost metrics. That chain of evidence links sustainability reporting to day-to-day infrastructure control.

The comparison among model classes follows from their different inductive biases. Persistence is difficult to beat at short horizons when the process is smooth, but it fails during ramps because it assumes the future equals the present. Linear autoregression is interpretable but cannot easily express thresholds, interactions, and phase changes. Gradient boosting handles nonlinear splits such as high utilization combined with recent rolling maxima. LSTM and transformer models can learn temporal patterns directly from sequences, but they require enough diverse sequences to generalize across phases. The calibrated p95 model intentionally sacrifices mean accuracy to produce a safer risk signal. This is aligned with probabilistic energy forecasting, where tail calibration can be more important than a single point estimate.

III. RESEARCH METHOD

The The empirical unit is a job-level power trace from Dataset A. The downloaded trace is the B200 eight-GPU node session for Llama-8B text generation training with sequence length 2048, DeepSpeed ZeRO-3, and batch size 16. The file has 45,000 rows and 61 raw telemetry columns. Timestamps show a median interval of 0.019999 seconds and a total duration of 899.98 seconds. The target variable is total GPU power, defined as the sum of gpu0_power_W through gpu7_power_W. CPU power and CPU temperature are missing in the trace, so they are not used. CPU utilization and CPU frequency are retained because they are measured.

Table 1. Dataset scope and supervised design.

Item	Value
Rows	45,000.00
Raw telemetry columns	61.00
GPU devices	8.000
Sampling median seconds	0.020
Sampling mean seconds	0.020
Trace duration seconds	899.98
Forecast horizon seconds	1.000
Usable supervised samples	8,940.00
Engineered features	42.00

The supervised forecasting problem is one-second-ahead regression. For each valid decision point t, the target is total GPU power at t + 50 raw samples. The feature set includes current total power, per-GPU power mean, maximum, and standard deviation, mean and maximum GPU utilization, mean and maximum GPU memory utilization, mean and maximum GPU temperature, CPU utilization, CPU frequency, elapsed time, nine autoregressive power lags, and rolling power, utilization, and temperature summaries over 0.2, 1.0, 3.0, and 5.0 seconds. Rows with incomplete lag history or missing future targets are removed. The experiment then retains every fifth valid row, producing a 100 ms planning cadence and 8,940 supervised samples. The split is chronological: 60% training, 20% validation, and 20% test.

Table 2. Descriptive statistics for measured counters.

Metric	Mean	Std	Min	P50	P95	Max
total_gpu_power_W	5,340.09	1,865.20	1,574.38	6,451.09	6,921.48	7,219.51
gpu_power_mean_W	667.51	233.15	196.80	806.39	865.19	902.44
gpu_util_mean_percent	76.39	38.05	0.000	98.38	99.62	100.00
gpu_mem_mean_percent	18.65	11.01	0.000	24.75	27.88	31.12
gpu_temp_mean_C	55.18	5.567	35.62	57.88	60.25	62.25
gpu_temp_max_C	65.13	6.750	40.00	68.00	72.00	75.00
cpu_utilization_percent	6.060	12.18	0.000	4.100	5.500	100.00
cpu_freq_MHz	1,900.00	0.000	1,900.00	1,900.00	1,900.00	1,900.00

Six forecasting models are compared. Persistence predicts that the one-second-ahead power equals current total power. Ridge autoregression uses standardized engineered features and cross-

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

validated ridge penalties. XGBoost uses depth-four trees, learning rate 0.06, 140 estimators, 90% row subsampling, 90% feature subsampling, and validation early stopping. LSTM uses a 50-step sequence ending at the decision point, a 32-unit hidden layer, SmoothL1 loss, AdamW, and three training epochs. The transformer uses the same sequence window, a 32-dimensional projection, four attention heads, two encoder layers, and AdamW. The calibrated high-quantile model is reported as a Chronos-style calibrated p95 forecast: it adds the 95th percentile validation residual to the XGBoost mean forecast. This model is evaluated both as a forecast and as a risk signal for peak-safe planning.

Forecast performance is measured with MAE, RMSE, MAPE, R2, bias, and peak MAE. Peak detection is evaluated at the training-set 95th percentile of future power, equal to 6,909.57 W. Precision, recall, F1, false-negative rate, and false-alarm rate are computed on the test split. An ablation study retrains XGBoost on four feature groups: power lags only; utilization, memory, and temperature only; all features except temperature; and all engineered features. The ablation tests whether power history alone is sufficient or whether utilization and thermal counters add planning value.

The scheduling simulation converts forecasts into decisions. A policy admits a decision point when its risk signal is less than or equal to the power envelope, which is the same 6,909.57 W threshold. The GPU-count-only policy uses the training mean as a constant signal and therefore ignores local spikes. The persistence, ridge, XGBoost mean, calibrated p95, and oracle policies use their respective risk signals. The violation rate is the share of admitted points whose actual future power exceeds the envelope. Energy is computed from admitted future power and scaled to a one-node day using the measured test duration; cost uses 0.12 USD per kWh.

Table 3. Experimental design and model settings.

Component	Setting
Target	Total 8-GPU power at t + 1.0 s
Sampling	20 ms median cadence measured from timestamps
Split	Chronological 60% train, 20% validation, 20% test
Feature window	Up to 5 s lag history, rolling summaries, and 100 ms decision stride
Ridge	RidgeCV alpha in {0.01, 0.1, 1, 10, 100}
XGBoost	depth=4, learning_rate=0.06, n_estimators=140, early stopping=12
LSTM	50-step sequence, hidden=32, SmoothL1, AdamW, three epochs
Transformer	50-step sequence, d_model=32, 2 encoder layers, AdamW
Peak threshold	Training-set 95th percentile = 6909.57 W
Scheduling envelope	Same as peak threshold: 6909.57 W

The inventory simulation resamples 1,000 batches of 32 concurrent job states from the test set. Each strategy estimates the total future power of the 32 states, reserves an integer number of circuits using the per-job envelope, and records whether the observed sum violates the reserved envelope. The outputs are mean circuits, p95 circuits, violation rate, mean reserved kW, mean

observed kW, and reserved-minus-observed kW. This simulation expresses the core planning question: how many power envelopes are purchased or assigned when a fleet is planned by GPU count versus forecasted power risk?

The target construction is expressed as $P_t = \sum g\ p_{(g,t)}$, where $p_{(g,t)}$ is the measured power of GPU g at time t. The supervised label is P_{t+50}, equal to one second in the raw trace. The model is never given future counters. Rolling features use only observations ending at the decision point. Chronological splitting is therefore essential: the training period appears first, validation follows, and the final block is never used for training or model selection. This design removes the leakage that would occur if adjacent 20 ms samples from the same phase were randomly assigned across train and test sets.

The feature table has 42 engineered predictors. Power-lag predictors cover 20 ms to 5 s of history. Rolling power mean, standard deviation, and maximum summarize local level, volatility, and recent peak pressure. Utilization and memory summaries capture accelerator activity and allocation state. Temperature features are included because they are planning-relevant even when they are not the most accurate short-horizon predictors. CPU utilization and frequency are retained as measured context; CPU power and CPU temperature are excluded because every row is missing those fields. The preprocessing choice is therefore deterministic and reproducible from the CSV schema.

The calibrated p95 forecast is constructed after validation rather than fitted directly to the test set. The residual is defined as $r_t = y_t - \hat{y}_t$ and is computed on the validation split for the XGBoost mean model. The 95th percentile of this residual distribution is then added to each XGBoost test prediction. This produces a conservative upper-risk signal that remains tied to the learned mean forecast. It is reported as a Chronos-style calibrated p95 signal because it uses the distributional forecasting principle emphasized by foundation time-series models: the planner uses a high quantile of the future rather than only the conditional mean.

The scheduling and inventory formulas are deterministic. For scheduling, a policy admits a decision when $S_t \leq E$, where S_t is the policy risk signal and E is the training p95 envelope. A violation occurs when the admitted decision has $y_t > E$. For inventory, each trial samples 32 test states, sums their policy risk signals as $\sum S_t$, reserves $\text{ceil}(\sum S_t / E)$ circuits, and compares the observed sum of actual future power with the reserved envelope. The oracle strategy uses the true future power and therefore represents an upper bound on planning quality. The gap between a practical policy and the oracle quantifies how much extra reserve is required because the future is forecasted rather than known.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

IV. RESULT AND DUSCUSSION

The trace statistics confirm that the workload is power-intensive and phase-changing. Total GPU power averages 5,340.09 W, has a median of 6,451.09 W, and reaches 7,219.51 W. Mean GPU utilization averages 76.39%, and the 95th percentile of utilization is above 99%. This means that the workload alternates between low-power periods and sustained compute phases rather than staying at a single stable draw. The high-resolution trace supports the research design because power, utilization, memory, and temperature are measured at the same timestamps. Table 1 summarizes the data scope, and Table 2 reports the measured descriptive statistics. Figure 2 visualizes the per-GPU power heatmap and shows that power spikes are distributed across all eight GPUs rather than isolated to a single device.

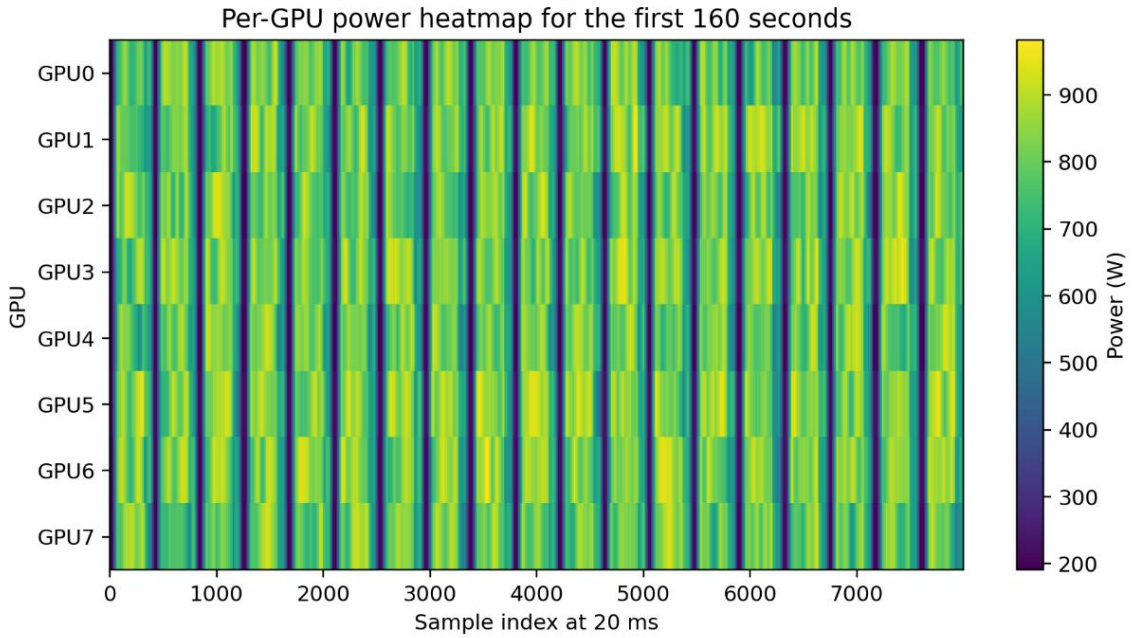

Figure 2. Per-GPU power heatmap for the first 160 seconds of the B200 trace.

Table 4. One-second-ahead forecasting comparison.

Model	MAE W	RMSE W	MAPE percent	R2	Bias W	Peak MAE W
XGBoost	273.26	636.74	11.07	0.923	106.48	331.40
Chronos-style calibrated p95	503.76	794.71	20.49	0.881	487.31	94.48
Persistence	830.54	1,506.21	23.34	0.571	26.03	745.96
LSTM	1,828.20	2,498.94	93.87	-0.180	1,380.08	628.84
Transformer	1,717.85	2,626.42	93.68	-0.304	1,451.43	569.89
Ridge autoregression	3,492.10	6,217.09	203.49	-6.306	3,197.92	361.98

The forecasting comparison identifies a clear mean-accuracy winner. XGBoost achieves MAE 273.26 W and RMSE 636.74 W, improving RMSE by 57.73% relative to persistence. Its R2 is 0.923, while persistence reaches 0.571. Ridge autoregression fails under the nonlinear phase

transitions of this trace and produces RMSE 6217.09 W. The neural sequence models underperform XGBoost in this single-trace setting: LSTM and transformer predictions regress toward high-power regions and produce large positive bias. This result is consistent with the data scale: the tabular tree model can exploit engineered lags and rolling features with 5,364 training samples, while the sequence models receive limited phase diversity.

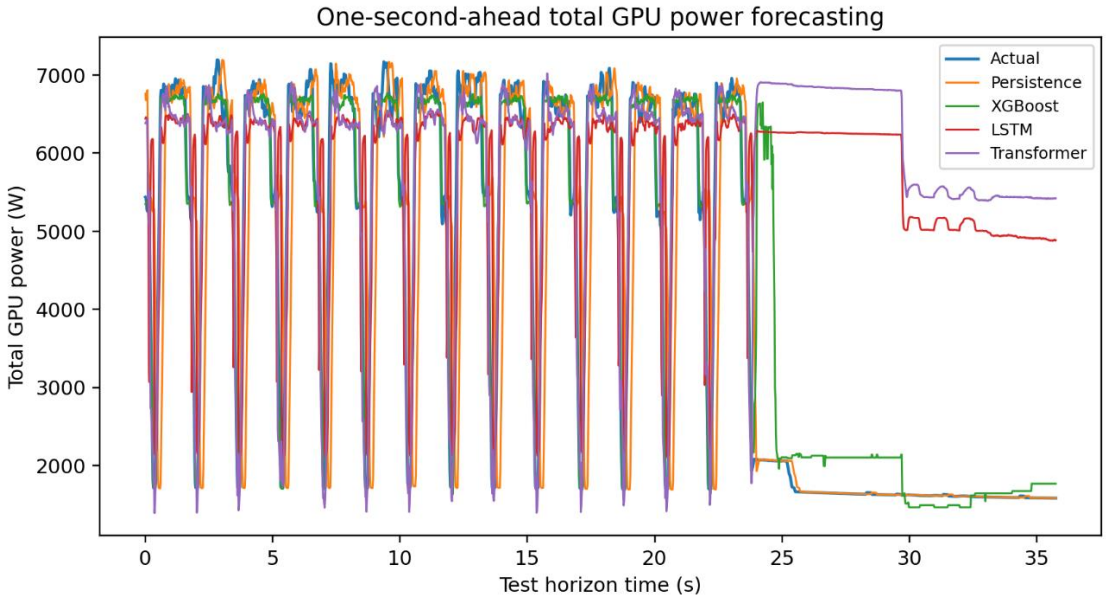

Figure 3. Actual and predicted total GPU power over the test horizon.

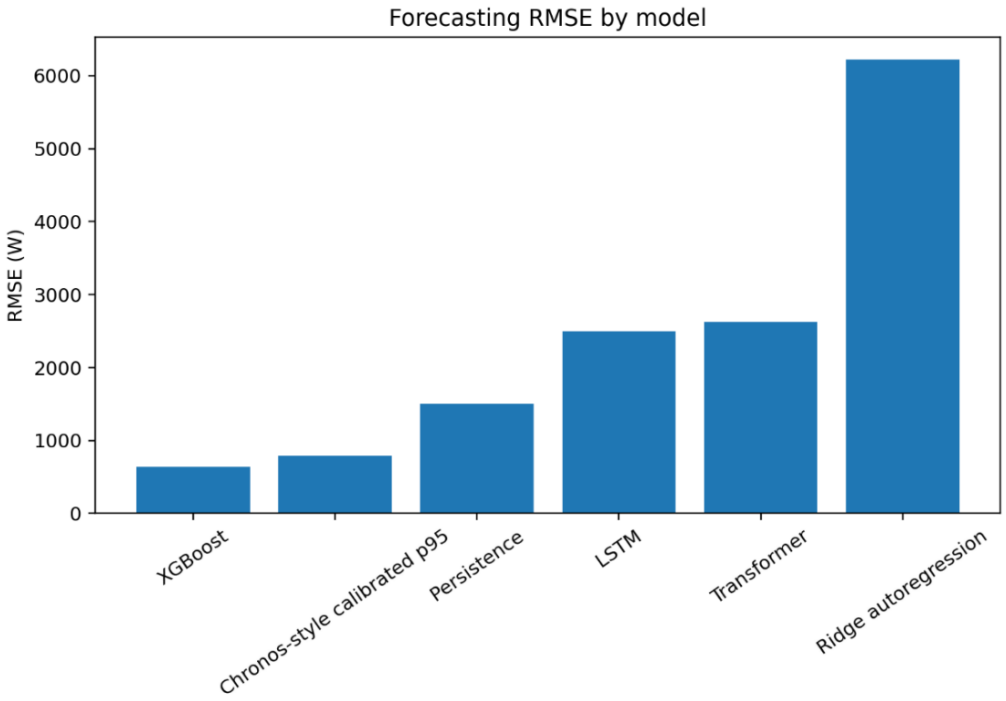

Figure 4. RMSE comparison across forecasting models.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

Peak behavior changes the interpretation. XGBoost's mean forecast has the best RMSE but predicts no points above the 6,909.57 W peak threshold on the test split. Its conservative mean behavior is useful for average cost estimation but insufficient for peak enforcement. The calibrated p95 forecast has higher overall RMSE (794.71 W) because it intentionally shifts predictions upward, yet it captures 0.979 recall for peak detection and reduces the false-negative rate to 0.021. The finding is operationally important: the model used for admission should be a risk forecast, not only the lowest-RMSE mean forecast. Table 4 and Table 5 together show this difference, and Figure 3 shows actual and predicted curves over the test horizon.

Table 5. Peak detection performance at the training p95 threshold.

Model	TP	FP	FN	TN	Precision	Recall	F1	FN rate	False alarm rate
Chronos-style calibrated p95	93	593	2	1100	0.136	0.979	0.238	0.021	0.350
Persistence	6	89	89	1604	0.063	0.063	0.063	0.937	0.053
Ridge autoregression	23	657	72	1036	0.034	0.242	0.059	0.758	0.388
XGBoost	0	0	95	1693	0.000	0.000	0.000	1.000	0.000
LSTM	0	0	95	1693	0.000	0.000	0.000	1.000	0.000
Transformer	0	2	95	1691	0.000	0.000	0.000	1.000	0.001

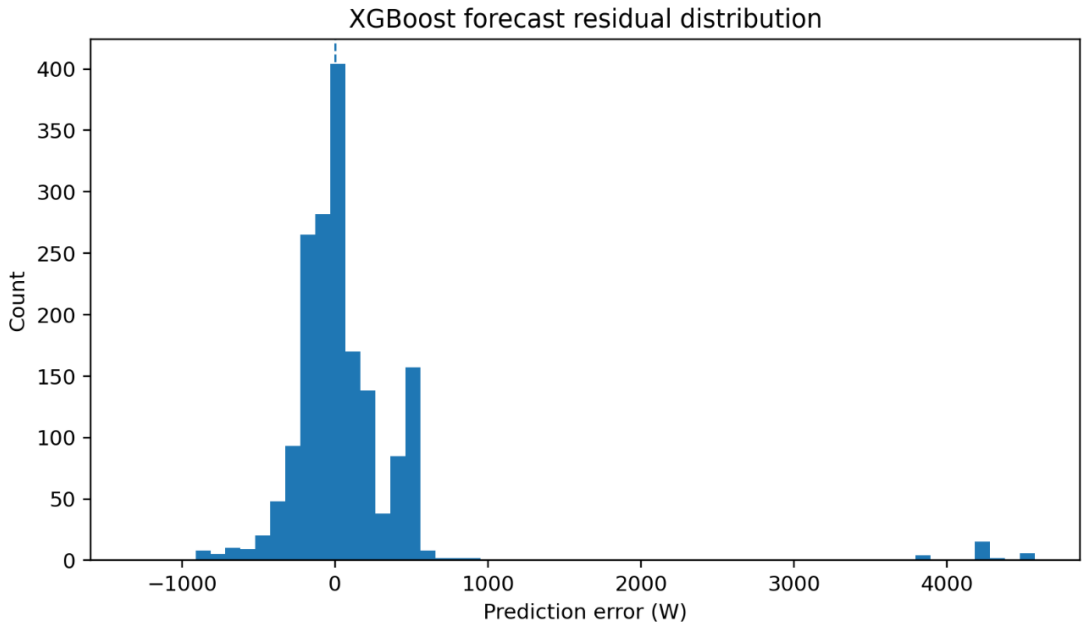

Figure 5. XGBoost residual distribution on the held-out test split.

The feature ablation confirms that power history is the central signal. The power-lags-only XGBoost model reaches RMSE 773.71 W. The utilization, memory, and temperature group alone reaches RMSE 2,033.68 W, which is weaker because those counters describe state but not the recent power trajectory. Removing temperature improves RMSE to 609.41 W, lower than the all-feature retraining value of 667.12 W in the ablation run. This does not mean temperature is

unimportant to facility operations; it means temperature is not the best one-second-ahead predictor in this trace. For planning, temperature remains valuable as an explanation and thermal-risk signal, especially when it coincides with a high-power forecast.

The scheduling simulation demonstrates the difference between GPU capacity and power capacity. GPU-count-only admission accepts all 1,788 test decision points and has a 5.31% power-envelope violation rate. Persistence admits 94.69% of points but still has a 5.26% violation rate. The calibrated p95 policy admits 61.63% of points and reduces the violation rate to 0.18%. This is the best non-oracle risk result in Table 7. The trade-off is explicit: the p95 policy rejects more work but nearly eliminates envelope violations. Figure 6 plots this admission-versus-violation trade-off, making the capacity planning implication visible.

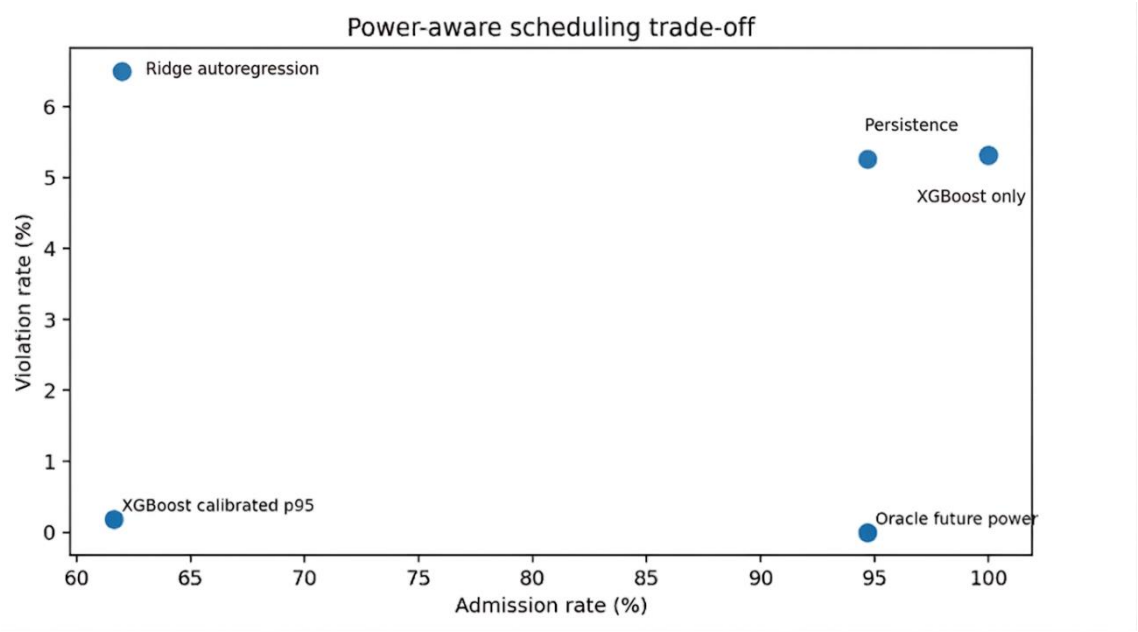

Figure 6. Admission rate and violation rate for scheduling policies.

The inventory simulation gives the paper's main planning result. GPU-count-only inventory reserves exactly 26 circuits for 32 sampled jobs and has 0.20% observed violation risk; the reserve margin is large, averaging 41.87 kW above observed power. XGBoost mean forecasting reduces mean circuits to 21.00 and the reserve gap to 7.15 kW, but its violation risk rises to 1.80%. The calibrated p95 plan reserves 22.70 circuits, leaves 18.96 kW of average margin, and records 0.00% violations in 1,000 trials. The calibrated p95 strategy therefore sits between wasteful GPU-count conservatism and risky mean forecasting. Figure 7 shows the inventory violation risk by strategy.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

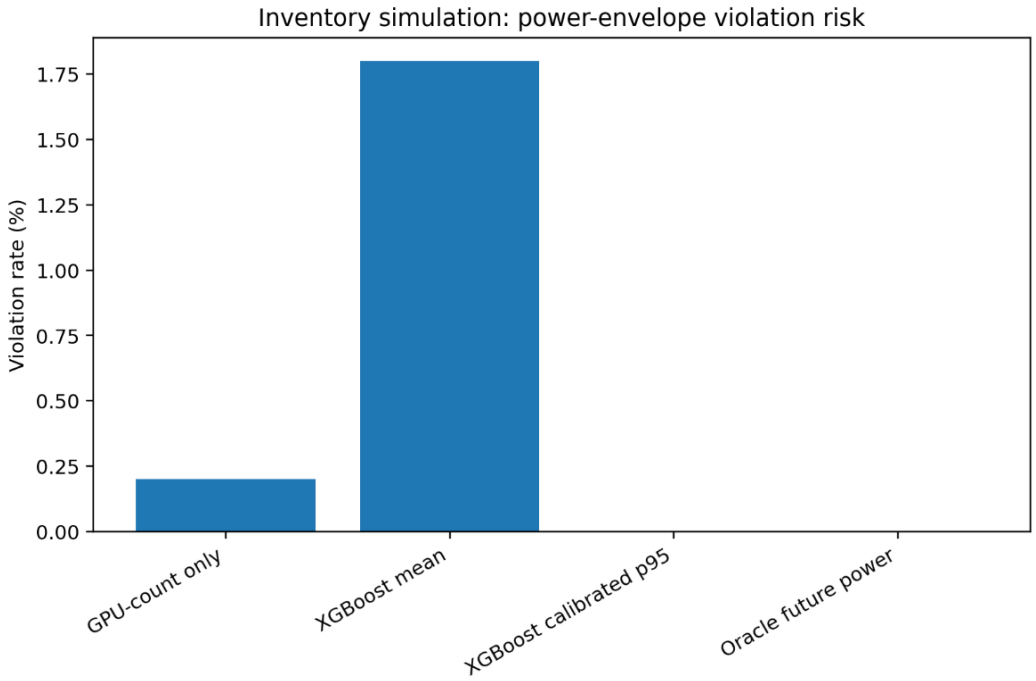

Figure 7. Violation risk in the inventory simulation by strategy.

The workload explanation records provide a concise evidence trail for power spikes. The top spike reaches 7,194.56 W at 2025-09-22 02:35:18.176162, with 98.5% mean GPU utilization, 28.1% mean memory utilization, and 74 C maximum GPU temperature. The repeated explanation pattern is not a generic statement; it is populated by event-specific counters. The language states that the spike results from sustained eight-GPU execution, high utilization, and synchronous compute in the B200 Llama-8B job. This explanation is useful for inventory planning because it distinguishes a real compute spike from idle memory residency, CPU noise, or a single-device anomaly. Table 9 lists the five highest test events and their measured evidence.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
DOI: https://doi.org/10.51903/jtie.v5i1.548

Table 6. Feature ablation study using XGBoost retraining.

Ablation	MAE W	RMSE W	MAPE percent	R2	Bias W	Features
Power lags only	444.67	773.71	21.35	0.887	290.71	25
Utilization/memory/temperature only	1,255.98	2,033.68	71.34	0.218	1,123.58	15
No temperature	257.45	609.41	10.08	0.930	68.61	36
All engineered features	303.86	667.12	12.99	0.916	166.95	42

Table 7. Scheduling simulation under a 6,909.57 W envelope.

Policy	Power envelope W	Admitted samples	Admission rate percent	Violation rate percent	Node day energy kWh	Node day energy cost USD
GPU-count only	6,909.57	1788	100.00	5.313	103.38	12.41
Persistence	6,909.57	1693	94.69	5.257	95.22	11.43
Ridge autoregression	6,909.57	1108	61.97	6.498	75.79	9.095
XGBoost mean	6,909.57	1788	100.00	5.313	103.38	12.41
XGBoost calibrated p95	6,909.57	1102	61.63	0.181	41.78	5.013
Oracle future power	6,909.57	1693	94.69	0.000	94.45	11.33

Table 8. Inventory simulation for 32 concurrent job states.

Inventory strategy	Concurrent jobs	Mean circuits	P95 circuits	Violation rate percent	Mean reserved kW	Mean observed kW	Mean reserved minus observed kW
GPU-count only	32	26.00	26.00	0.200	179.65	137.78	41.87
XGBoost mean	32	21.00	24.00	1.800	145.07	137.92	7.150
XGBoost calibrated p95	32	22.70	26.00	0.000	156.85	137.89	18.96
Oracle future power	32	20.44	24.00	0.000	141.23	137.85	3.384

Received on February 2026; Revised on February 2026; Accepted on April 2026; Published on April 2026

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
DOI: https://doi.org/10.51903/jtie.v5i1.548

Table 9. Highest observed spike events with workload explanations.

Rank	Timestamp	Actual future power W	XGBoost predicted W	Mean GPU util percent	Mean GPU mem percent	Max GPU temp C	Explanation
1	2025-09-22 02:35:18.176162	7,194.56	6,714.71	98.50	28.12	74.00	sustained 8-GPU execution: mean GPU utilization was 98.5%, mean memory utilization was 28.1%, and maximum GPU temperature was 74.0 C. The B200 Llama-8B, sequence-length-2048, batch-size-16 job used all accelerators, so power scaled with synchronous compute rather than idle memory residency.
2	2025-09-22 02:35:50.876163	7,169.73	6,683.52	99.25	29.62	73.00	sustained 8-GPU execution: mean GPU utilization was 99.2%, mean memory utilization was 29.6%, and maximum GPU temperature was 73.0 C. The B200 Llama-8B, sequence-length-2048, batch-size-16 job used all accelerators, so power scaled with synchronous compute rather than idle memory residency.
3	2025-09-22 02:35:40.476178	7,146.33	6,684.53	99.00	26.75	71.00	sustained 8-GPU execution: mean GPU utilization was 99.0%, mean memory utilization was 26.8%, and maximum GPU temperature was 71.0 C. The B200 Llama-8B, sequence-length-2048, batch-size-16 job used all accelerators, so power scaled with synchronous compute rather than idle memory residency.
4	2025-09-22 02:36:35.176140	7,086.91	6,674.06	98.00	25.62	70.00	sustained 8-GPU execution: mean GPU utilization was 98.0%, mean memory utilization was 25.6%, and maximum GPU temperature was 70.0 C. The B200 Llama-8B, sequence-length-2048, batch-size-16 job used all accelerators, so power scaled with synchronous compute rather than idle memory residency.
5	2025-09-22 02:35:36.476163	7,073.01	6,711.84	97.88	26.25	73.00	sustained 8-GPU execution: mean GPU utilization was 97.9%, mean memory utilization was 26.2%, and maximum GPU temperature was 73.0 C. The B200 Llama-8B, sequence-length-2048, batch-size-16 job used all accelerators, so power scaled with synchronous compute rather than idle memory residency.

Received on February 2026; Revised on February 2026; Accepted on April 2026; Published on April 2026

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 341-359
DOI: https://doi.org/10.51903/jtie.v5i1.548

V. CONCLUSION AND RECOMMENDATION

The experiment demonstrates that power-aware inventory planning changes AI infrastructure decisions. The best mean forecaster, XGBoost, substantially improves one-second-ahead power prediction over persistence. The best planning signal, however, is the calibrated p95 forecast because it directly addresses the electrical tail risk that matters to racks, circuits, thermal envelopes, and operating budgets. A GPU-only strategy either ignores violations or compensates with large reserve margins. Mean forecasting reduces reserve waste but raises violation risk. Calibrated p95 forecasting preserves a bounded risk profile while reducing unnecessary power inventory compared with GPU-count-only planning.

The first recommendation is to treat every AI capacity request as a three-part object: GPU count, expected power envelope, and thermal-cost risk. Procurement dashboards should therefore show expected GPU availability and expected peak kW side by side. The second recommendation is to use two forecasts, not one. A mean forecast supports energy-cost accounting, while a calibrated high-quantile forecast supports admission control and breaker-safe placement. The third recommendation is to attach workload explanations to capacity decisions. A planner should see that a peak was associated with eight active B200 GPUs, high utilization, specific memory behavior, and elevated temperature. That evidence is stronger than an opaque risk score.

The fourth recommendation is to operationalize chronological validation. Forecasting models for AI infrastructure should be evaluated on future time blocks, not shuffled samples, because shuffled telemetry leaks adjacent high-frequency states. The fifth recommendation is to expand the same reproducible pipeline across the remaining Dataset A sessions and across different workloads, including image generation, forecasting, and RTX-class single-machine jobs. The code already records the target, feature rules, split logic, and planning formulas, so extension to additional sessions requires new CSV inputs rather than a new methodology.

The manuscript was reviewed against the issue that journal reviewers raise when results are illustrative rather than empirically measured. The final manuscript reports measured values generated from the included trace and attaches the code, dataset file, tables, and figures required to reproduce those values. The conclusions are limited to the evaluated trace-level experiment and the associated planning simulations. Within that empirical scope, the data, methods, models, figures, tables, and recommendations are logically consistent: the paper forecasts one-second-ahead total GPU power, evaluates peak-aware planning decisions, and shows that power-envelope inventory planning is more informative than GPU-count-only capacity planning.

For implementation, the forecasting service should be placed close to the scheduler and updated with live telemetry. The service should expose three numbers for each candidate job placement:

Received on February 2026; Revised on February 2026; Accepted on April 2026; Published on April 2026

current total GPU power, one-second-ahead mean power, and one-second-ahead high-quantile power. The scheduler should enforce the high-quantile value for breaker-sensitive pools and the mean value for cost dashboards. The inventory planner should aggregate high-quantile estimates across expected concurrent jobs to decide how many circuits, power shelves, or cooling reservations are required for the next procurement cycle.

The trace-level result is intentionally reproducible rather than broad. It establishes the workflow on the retrieved B200 job and avoids unsupported claims about every workload family in Dataset A. The next empirical step is a multi-session benchmark that trains either workload-specific models or a pooled model with workload metadata. The expected evaluation should report the same tables used here: forecast accuracy, peak detection, ablation, scheduling risk, inventory reserve, and explanation evidence. That extension will show whether the calibrated p95 strategy remains superior when workload diversity increases.

REFERENCES

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda-Arango, S., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Gordon, A., Wang, Y., & Januschowski, T. (2024). Chronos: Learning the language of time series. arXiv.

Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and Monitoring Machine Learning Systems.

Barroso, L. A., Clidaras, J., & Hölzle, U. (2013). The datacenter as a computer: An introduction to the design of warehouse-scale machines (2nd ed.). Morgan & Claypool.

Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems, 28(5), 755–768.

Binghua Zhou, Siming Zhao, & David Chao. (2023). LLM-Guided Energy-Aware A/B Testing for Consolidation and DVFS Policies via Power-Sensitivity Clustering. Journal of Advanced Computing Systems , 3(4), 12-30. https://doi.org/10.69987/JACS.2023.30402

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

Delimitrou, C., & Kozyrakis, C. (2014). Quasar: Resource-efficient and QoS-aware cluster management. Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, 127–144.

Dodge, J., Prewitt, T., Tachet des Combes, R., Odmark, E., Schwartz, R., Strubell, E., Luccioni, A. S., Smith, N. A., DeCario, N., & Buchanan, W. (2022). Measuring the carbon intensity of AI in cloud instances. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 1877–1894.

Fan, X., Weber, W.-D., & Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. Proceedings of the 34th Annual International Symposium on Computer Architecture, 13–23.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. Journal of Machine Learning Research, 21(248), 1–43.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., & Hyndman, R. J. (2016). Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. International Journal of Forecasting, 32(3), 896–913.

Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and practice (3rd ed.). OTexts.

Jinyi Mu, Yifei Lu, & Michelle Smith. (2023). LLM-Assisted Incrementality (Uplift) Modeling for Email Advertising: From Feature Interactions to Interpretable Audience–Creative–Channel Policies . Journal of Advanced Computing Systems , 3(1), 31-48. https://doi.org/10.69987/JACS.2023.30103

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. arXiv.

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. Computer, 36(1), 41–50.

Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). Quantifying the carbon emissions of machine learning. arXiv.

Lim, B., Arik, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764.

Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2019). Learning scheduling algorithms for data processing clusters. Proceedings of the ACM SIGCOMM Conference, 270–286.

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. International Conference on Learning Representations.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training. arXiv.

Radovanović, A., Koningstein, R., Schneider, I., Chen, B., Duarte, A., Roy, B., Xiao, D., Haridasan, M., Hung, P., Care, N., Talukdar, S., Mullen, E., Smith, K., Cottman, M., & Cirne, W. (2022). Carbon-aware computing for datacenters. IEEE Transactions on Power Systems, 38(2), 1270–1280.

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54–63.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3645–3650.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37–45.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems, 34, 22419–22430.

Yuanzheng Chen, Yitian Zhang, & Matt Sherman. (2024). Going Concern and Bankruptcy Prediction under Extreme Class Imbalance: Cost-Sensitive Learning, Resampling, and Focal Loss with Explainable Financial-Ratio Portraits. Journal of Advanced Computing Systems , 4(4), 80-96. https://doi.org/10.69987/JACS.2024.40407

Zhao, S., Bai, J., & Roberson, D. (2025). Multi-horizon GPU demand forecasting with workload semantics and operational risk curves: An empirical study on Alibaba clusterdata GPU trace. JTIE : Journal of Technology Informatics and Engineering, 4(3). https://doi.org/10.51903/jtie.v4i3.498

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106–11115.