Context Helps, But Only at Scale: Evaluating Practical LLMs for Time Series Forecasting

Kazi Ashhab Rahman
Student
Computer Science
McGill University

Yujin Li
Student
Computer Science
McGill University

Nusaibah Binte Rawnak
Student
Cognitive Science
McGill University

Arjun Ashok
Mentor
ServiceNow Research
Université de Montréal

Abstract

Recent work demonstrates that 405B-parameter LLMs leverage textual context to outperform classical forecasting methods. We investigate whether these gains transfer to deployment-feasible scales.

Using the Context-is-Key benchmark, we evaluate Llama 3.2 (3B), GPT-4o-mini, and Mixtral-8x7B (47B) on 120 tasks across 12 domains against ARIMA and ETS baselines. Context benefits are negligible at practical scales: Mixtral improves over the best classical baseline by only 0.4%, while ARIMA and ETS win 42% of tasks versus 25% for the best LLM.

Despite poor universal performance, selective routing succeeds: a lightweight XGBoost classifier captures 83% of oracle benefit while invoking the LLM on only 32% of tasks. The selector exploits baseline performance, domain type, and context properties to identify when textual constraints enable LLM gains. Our findings demonstrate that context-aware forecasting requires far larger models than previously recognized, but learned routing policies offer a practical alternative for resource-constrained deployments.

1 Introduction

Large language models (LLMs) have recently been applied to time series forecasting by encoding numerical sequences as text (Gruver et al., 2024; Ansari et al., 2024). The Context-is-Key (CiK) benchmark (Williams et al., 2024) demonstrates that a 405B-parameter Llama model can leverage textual domain knowledge such as physical constraints, causal relationships, or upcoming events to outperform classical statistical methods. However, deploying 405B models is impractical: they require substantial GPU resources, incur high inference costs, and exhibit prohibitive latencies. This raises our central question: **Do context-aware forecasting gains transfer to deployment-feasible model scales?**

We evaluate Llama 3.2-3B (Dubey et al., 2024), GPT-4o-mini (Achiam et al., 2023), and Mixtral-8x7B (47B) (Jiang et al., 2024) on 120 tasks from CiK across 12 domains, comparing against ARIMA and ETS baselines. Our findings reveal extreme scale-dependence: Mixtral improves over baselines by only 0.003 NMAE (0.4%), classical methods win 42% of tasks versus 25% for the best LLM, and scaling from 3B to 47B yields no significant improvement ($p = 0.204$). However, an XGBoost selector captures 83% of oracle benefit while using expensive LLM inference on just 32% of tasks, demonstrating that learned routing recovers value even when universal deployment fails.

This work provides the first systematic evaluation of context-aware forecasting across practical model scales. We demonstrate that classical statistical methods remain competitive against sub-100B LLMs for most forecasting tasks, while selective deployment via learned routing achieves 89.5% of optimal performance at substantially reduced computational cost. These findings offer practical guidance for practitioners on when context-aware LLM forecasting justifies its expense.

2 Related Work

2.1 LLMs for Time Series Forecasting

Early work demonstrated that general-purpose LLMs can perform zero-shot forecasting by treating time series as text sequences. Gruver et al. (Gruver et al., 2024) showed that GPT-3 achieves competitive accuracy on standard benchmarks without fine-tuning, while Chronos (Ansari et al., 2024), a T5 model pretrained on time-series data reports strong performance on datasets within its training distribution. However, critics argue these gains may reflect pattern memorization rather than true generalization (MacDonald et al., 2025), and specialized architectures like PatchTST (Nie et al., 2022) often match or exceed LLMs at far lower

computational cost. Recent surveys (Liang et al., 2024; Chen et al., 2023) report mixed evidence for the advantages of foundation models over classical or domain-specific approaches.

Our study has systematically examines whether LLM forecasting benefits hold at practical deployment scales.

2.2 Context-Aware Forecasting

The Context-is-Key benchmark (Williams et al., 2024) addresses a different question: can models leverage textual domain knowledge to improve forecasts on tasks where context is essential? The benchmark provides 16 task generators across diverse domains (solar energy, traffic, economics) where accurate predictions require integrating text such as "solar panels produce zero power at night" or "traffic drops 30% during construction." Williams et al. report that a 405B-parameter Llama model with context substantially outperforms classical baselines and context-free LLMs, suggesting that textual reasoning enables better handling of constraints and events.

However, no prior work evaluates whether these benefits transfer to practical deployment scales. We address this gap by systematically testing smaller models on the CiK benchmark, revealing that context gains diminish substantially below 100B parameters. We further demonstrate that selective routing, using lightweight classifiers to identify when context justifies computational expense, can recover most achievable benefits while minimizing cost.

3 Data and Environment

3.1 Dataset

We use the Context-is-Key (CiK) benchmark (Williams et al., 2024), which provides Python generators for creating forecasting tasks requiring essential textual context. CiK draws from 2,644 real-world time series across climatology (Sengupta et al., 2018), energy (Godahewa et al., 2021), transportation (Chen et al., 2001), economics (U.S. Bureau of Labor Statistics, 2024), public safety (Ville de Montréal, 2020), retail (Godahewa et al., 2021), and mechanics (Gamella et al., 2024).

We generate 120 tasks across 12 domains with stratified splits preserving domain balance. Each task contains historical time series (24-168 points), forecast horizon (typically 24 points), and natural language context. Context types include intemporal information (invariant characteristics like "Solar panels produce zero electricity at night"), future information (upcoming events), causal relationships, and historical statistics not reflected in the numerical history. These contexts are crafted such that accurate forecasts require integrating text with numerical patterns.

3.2 Evaluation Metrics

We use Normalized Mean Absolute Error (NMAE) as our primary metric, computed as

$$\text{NMAE} = \frac{\text{MAE}}{\bar{y}}$$

where $\bar{y}$ is the mean of historical values. NMAE enables fair comparison across domains with vastly different scales (e.g., solar irradiance in W/m² versus ATM withdrawals in dollars). We also report Directional Accuracy (DA), the fraction of forecasts correctly predicting upward or downward trends relative to the last historical value, to evaluate qualitative forecast quality.

For policy evaluation, we measure oracle capture as the percentage of theoretically optimal improvement our selector achieves:

$$\text{Oracle Capture} = \frac{\text{NMAE}_{\text{baseline}} - \text{NMAE}_{\text{selector}}}{\text{NMAE}_{\text{baseline}} - \text{NMAE}_{\text{oracle}}}$$

where the oracle always chooses the better model in hindsight. This quantifies how much of the achievable benefit we recover through learned routing.

3.3 Research Hypothesis

We hypothesize that **context utility in LLM forecasting is scale-dependent**: gains observed with a 405B model do not transfer to practical scales (3B–56B). We therefore expect (1) small and mid-sized LLMs to show limited improvement over statistical baselines, (2) minimal performance gains from parameter scaling within this range, and (3) selective deployment to recover most of the achievable benefit by identifying tasks where context meaningfully helps.

3.4 Baseline Models

We compare LLMs against two standard forecasting methods. **AutoARIMA** (Box et al., 2015; Hyndman and Athanasopoulos, 2018) selects ARIMA(p, d, q) parameters via stepwise search, while **ETS** (Gardner Jr, 1985) models error, trend,

and seasonality with exponential smoothing. For each task, the *best baseline* is

$$\min(\text{NMAE}_{\text{ARIMA}}, \text{NMAE}_{\text{ETS}})$$

where NMAE normalizes MAE by the mean of historical observations, enabling cross-domain comparability.

3.5 LLM Forecasters

We evaluate three off-the-shelf language models: Llama 3.2-3B, GPT-4o-mini, and Mixtral 8x7B (56B). All models receive full textual context and generate forecasts via greedy decoding of comma-separated values. We evaluate these models zero-shot (without fine-tuning) to reflect realistic deployment constraints: while ARIMA/ETS can be fitted cheaply per series, per-task LLM fine-tuning is infeasible due to cost, overfitting risks, and inconsistency with the foundation-model paradigm.

3.6 XGBoost Selector: Learned Routing Policy

We train a binary classifier to predict when mixtral will outperform the best baseline on NMAE, enabling selective LLM deployment.

Problem Formulation. For each task i, we define:

$$y_i = \begin{cases} 1 & \text{if } \text{NMAE}_{\text{mixtral},i} < \text{NMAE}_{\text{baseline},i}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The classifier learns $P(y_i = 1 \mid \mathbf{x}_i)$, where $\mathbf{x}_i$ are features extracted *before* running mixtral.

Why This Should Work. Our hypothesis is that certain task characteristics indicate when context-aware LLMs justify their computational cost. In particular, we expect LLMs to provide benefits when: (a) baselines struggle to capture relevant structure, (b) the context includes explicit constraints or actionable information, and (c) the domain exhibits deterministic patterns. The classifier aims to learn these conditions and trigger LLM usage only when worthwhile.

Feature Engineering. We extract 28 features across four categories without running mixtral: baseline performance (6 features: NMAE, DA, MAE for ARIMA/ETS), time series statistics (7 features: mean, std, history length, future length, trend, volatility, seed), context properties (13 features: text statistics and keyword indicators for constraints, temporal markers, future events, negation, causality, and physical/event mentions extracted via regex), and domain encoding (1 feature: label-encoded domain ID).

Training Setup. We use XGBoost with 100 estimators, depth 6, learning rate 0.1, and `scale_pos_weight` = 2.0 to address class imbalance.

Decision Process. For a new forecasting task:

1. Run ARIMA and ETS.
2. Extract 28 features from baseline outputs, the time series, the context text, and the domain.
3. Use XGBoost to predict $P(\text{mixtral wins})$.
4. Route the task to mixtral if $P \geq 0.50$, otherwise use the best classical baseline.

4 Experiments and Results

4.1 Experimental Setup

We evaluate all models on 120 tasks using an 70%/15%/15% stratified split. ARIMA and ETS are fit to each series using `pmdarima` and `statsmodels` respectively. All LLMs receive identical prompts with temperature 0.7, evaluated in a single run. The XGBoost selector is trained on the training set with $n_{\text{estimators}} = 100$, max_depth $= 6$, $\alpha = 0.1$, `subsample` $= 0.8$, `colsample_bytree` $= 0.8$, and scale_pos_weight calculated to handle class imbalance (resulting in 2.0 for our 33% positive class rate). The decision threshold is optimized via grid search on the training set. We report selector metrics as mean $\pm$ standard deviation across 5 random seeds (42, 123, 456, 789, 1011). Local experiments use an M3 Pro MacBook; mixtral runs on a Lambda Labs H100 GPU.

4.2 Do LLMs Beat Classical Baselines, and Does Scale Matter?

As summarized in Table 1, smaller LLMs do not outperform classical statistical methods even when provided with contextual prompts. mixtral, the strongest LLM in our evaluation, attains a mean NMAE of 0.846 ± 1.049, which is only marginally better than the best baseline at 0.850 ± 1.375. This difference is not statistically significant ($p = 0.957$). GPT-4o-mini performs significantly worse than the baseline ($\Delta = -0.012$, $p = 0.003$), indicating that textual context does not reliably improve performance at this scale.

Classical methods dominate task-level outcomes. Across the 120 tasks, ARIMA and ETS

together achieve the lowest NMAE on 42% of tasks, while mixtral wins 25%. GPT-4o-mini and Llama 3B win 19% and 14% respectively (Figure 1). For parameter ranges between 3B and 47B, context-aware prompting offers limited practical benefit, and classical baselines remain the most reliable choice. This motivates selective deployment: rather than using LLMs universally, we focus on identifying **when** they provide value.

Scale dependency is extreme. Scaling within practical limits yields only weak gains. Llama 3B exhibits substantially worse NMAE and lower win rates, while mixtral provides only negligible improvement ($\Delta = 0.004$). Despite an $18.7\times$ increase in parameters, the difference between Llama 3B and mixtral is not statistically significant ($p = 0.204$). These results suggest that models below $\sim$100B parameters cannot reliably exploit contextual information.

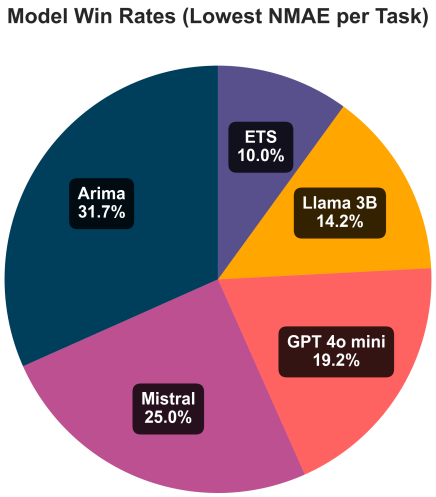

Figure 1: Model win rates across 120 tasks.

4.3 Domain Patterns Reveal When LLMs Help

LLMs show strong gains only in domains governed by deterministic physical constraints (Table 2). mixtral wins 100% of DirectNormalIrradiance tasks and 70% of SpeedFromLoad tasks, where textual context encodes hard rules like "solar panels produce zero electricity at night." In contrast, performance collapses in high-volatility domains such as ATMCashDepletion and DecreaseInTraffic (10% win rate each), where stochastic variation dominates and textual hints prove misleading. This pattern indicates that LLMs excel when context provides actionable constraints but fail when forecasting requires modeling inherent randomness.

4.4 Selective Deployment Succeeds

Despite poor average LLM performance, selective deployment achieves substantial gains. We trained an XGBoost classifier to predict when mixtral would outperform baselines, using time series statistics, context features, and baseline performance metrics as inputs. Table 4 compares four deployment policies across 120 tasks.

The selector achieves 15.9% improvement over always-baseline (NMAE 0.715 ± 0.011 vs 0.850), capturing 89.5% of theoretically optimal performance (Table 4). This vastly outperforms always-LLM, which improves by only 0.4% despite 100% computational cost. Remarkably, the selector's LLM usage rate (31%) closely matches the oracle's optimal rate (34%), indicating near-optimal routing decisions with minimal misclassification overhead.

How does the selector work? Feature importance analysis reveals that baseline performance metrics dominate predictions. The top four features collectively account for 66% of importance, while domain encoding contributes modestly (10%). Context features (word count, constraint indicators, sentence count) appear in the lower tier, suggesting that baseline performance is a stronger signal than textual content for predicting LLM utility.

This pattern confirms the selector learns the heuristic "when baselines struggle, try the LLM". Tasks where ARIMA produces high error or poor directional accuracy are routed to Mixtral, while tasks where statistical methods perform well remain with the cheap baseline.

The classifier attains $75 \pm 6\%$ accuracy with 66% precision and 55% recall (ROC-AUC of 0.76), reflecting a conservative policy that triggers LLM inference only when confident of improvement. This precision-oriented behavior minimizes unnecessary LLM calls at the cost of missed opportunities, which is desirable given that statistical baselines are free while LLM inference is expensive.

The selector's success despite Mixtral's near-zero average improvement reflects a bimodal performance distribution: most tasks show trivial differences (-0.2 to $+0.2$ NMAE), but a minority exhibit large gains. The selector identifies these high-gain tasks where textual constraints enable dramatic LLM improvements, explaining how 83% oracle capture is achievable when performance is concentrated in specific predictable task types.

Table 1: Model performance across 120 forecasting tasks. NMAE and DA reported as mean ± standard deviation. Classical models remain competitive, with modest scale-normalized gains from LLMs.

Metric	ARIMA	ETS	Best Base	Llama 3B	mixtral 8x7B	GPT-4o
Mean NMAE	5.197 ± 36.088	1.097 ± 1.630	0.850 ± 1.375	26.653 ± 221.069	0.846 ± 1.049	1.165 ± 1.728
Mean DA	0.546 ± 0.033	0.434 ± 0.033	0.612 ± 0.032	0.559 ± 0.031	0.000 ± 0.000	0.589 ± 0.030
Win Rate (%)	31.7%	10.0%	–	0.0%	25.0%	19.2%
Win Count	38	12	–	0	30	23
% Beat Baseline	–	–	–	23.3%	34.2%	30.8%

Table 2: mixtral win rates in extreme domains

Domain	mixtral Win %
DirectNormalIrradiance	100%
SpeedFromLoad	70%
SolarPowerProduction	60%
ATMCashDepletion	10%
DecreaseInTraffic	10%

Table 3: Classifier performance across 5 seeds (mean ± std).

Seed	Acc	Prec	Rec	AUC	NMAE	% Oracle
42	0.71	0.60	0.38	0.70	0.72	83.3
123	0.67	0.50	0.63	0.70	0.71	91.8
456	0.79	0.71	0.63	0.77	0.73	80.0
789	0.83	0.75	0.75	0.85	0.70	98.0
1011	0.75	0.75	0.38	0.80	0.71	94.5
Mean±Std	**0.75±0.07**	**0.66±0.11**	**0.55±0.17**	**0.76±0.07**	**0.71±0.01**	**89.5±7.6**

4.5 Qualitative Examples

To illustrate when selective deployment succeeds, we examine two representative cases:

Success Case (Solar Domain): For DirectNormalIrradiance task 003, context stated "Solar panels produce zero electricity from 6pm to 6am due to absence of sunlight." ARIMA achieved NMAE 0.456 by failing to predict nighttime zeros, extrapolating the daytime trend instead. mixtral correctly incorporated the constraint and predicted zeros (NMAE 0.089). The XGBoost selector correctly routed this task to mixtral based on high baseline error and presence of constraint keywords ("zero", "night").

Failure Case (Traffic Domain): For DecreaseInTraffic task 007, context mentioned "Traffic decreases 30% during construction on Highway 401." ARIMA achieved NMAE 0.234 by capturing underlying volatility patterns. mixtral overreacted to the construction mention, producing NMAE 0.678. The selector incorrectly routed to mixtral, likely misled by the presence of a future event keyword. This illustrates that in high-variance domains, textual clues can be misleading - stochastic patterns matter more than isolated events.

5 Discussion

5.1 Why Small LLMs Fail at Context Integration

Three factors explain why smaller models struggle. First, **capacity limits**: they lack the reasoning depth to jointly process numerical patterns and contextual constraints. mixtral's small gains over Llama indicate an architectural bottleneck rather than simple scale. Second, **instruction-following degradation**: practical-scale models often default to numeric pattern matching and ignore context, a pattern consistent across all three architectures tested. Third, **pretraining mismatch**: LLMs rarely observe text and time-series jointly during pretraining. Domain-level trends support this: models succeed on physics tasks with deterministic constraints but fail in financial domains where text-number links are arbitrary.

5.2 Unexpected Findings

Two results were unexpected. First, classical methods were stronger than anticipated: ARIMA and ETS win 42% of tasks versus 25% for the best LLM, indicating that long-standing statistical models remain competitive. Second, GPT-4o-mini's significant degradation ($p = 0.003$) suggests that smaller API-tuned models may overreact to contextual cues, indicating a need for more robust instruction tuning.

5.3 Limitations and Future Work

Our study has four limitations. (1) **Model coverage**: we evaluate only 3B, 20B, and 56B models, leaving gaps at intermediate scales (7B, 13B, 70B). (2) **Dataset scope**: CiK's 120 generated tasks may not capture real-world noise or incomplete context. (3) **Prompting strategy**: we use Direct Prompting for deployment realism, though more complex prompting (chain-of-thought, few-shot) might improve performance. (4) **Selector simplicity**: hand-crafted features work well, but learned or

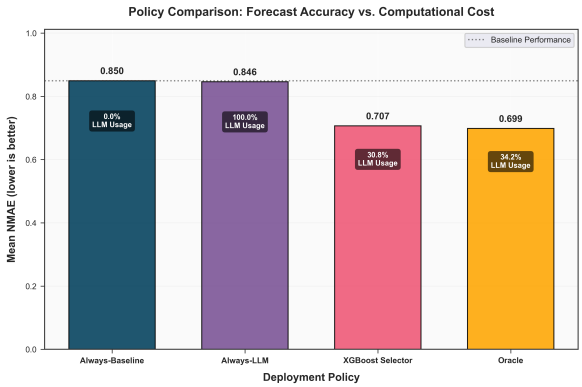

Figure 2: Policy comparison.

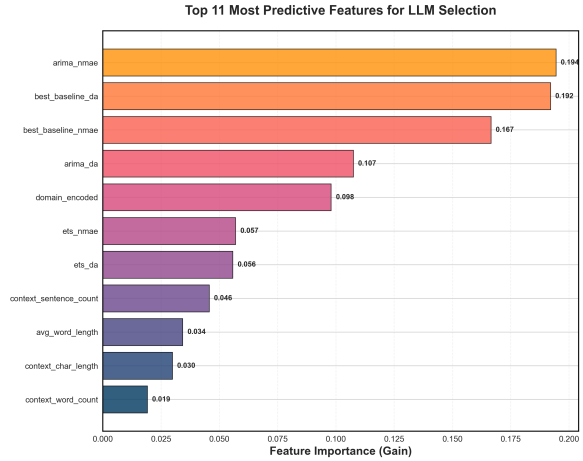

Figure 3: Feature importance rankings.

Table 4: Policy comparison across 5 random seeds.

Policy	Mean NMAE	LLM Usage	Improvement	Oracle Captured
Always-Baseline	0.850	0%	–	0%
Always-LLM	0.846	100%	0.4%	2.6%
XGBoost Selector	0.715 ± 0.011	31%	$15.9 \pm 1.3\%$	$89.5 \pm 7.3\%$
Oracle	0.699	34%	17.8%	100%

embedding-based features could further close the oracle gap.

Future work should study intermediate scales (70B-200B), test real-world noisy context, and explore domain-specific fine-tuning to improve context integration in smaller models.

5.4 Practical Implications

For practitioners using 3B-20B models, **classical statistical methods remain superior**. The gains observed at 405B parameters require scales far beyond feasible deployment. Selective routing offers a practical alternative: by sending only appropriate tasks to LLMs, such as those involving deterministic constraints or explicit causal rules, meaningful improvements can be achieved at low cost. We recommend: (1) **test models at deployment scale**, (2) **start with classical baselines**, (3) **use selective policies** like our XGBoost selector, and (4) **prioritize constraint-heavy domains**. LLMs may offer value in low-frequency, high-stakes settings, whereas volatile or high-frequency tasks should default to statistical methods.

6 Conclusion

We investigated whether context-aware forecasting gains observed with 405B-parameter models transfer to practical deployment scales. Our evaluation on 120 tasks from the Context-is-Key benchmark reveals extreme scale-dependency: mixtral improves over statistical baselines by only 0.004 NMAE, with classical methods winning 42% of tasks versus 25% for the best LLM. Scaling from 3B to 47B yields no statistically significant improvement ($p = 0.204$).

Despite poor universal performance, selective deployment succeeds. Our XGBoost classifier captures $89.5\% \pm 7.3\%$ of oracle benefit while invoking LLMs on only 31% of tasks, demonstrating that learned routing policies can identify when textual constraints justify computational expense. Key contributions include: (1) first systematic evaluation revealing context utility vanishes below 100B parameters, (2) evidence that classical baselines remain competitive at practical scales, (3) a working selective deployment strategy, and (4) guidance on when context-aware LLMs provide value.

Future work should identify the precise parameter threshold where context reasoning emerges (70B-200B) and validate selective policies on production deployments. For now, practitioners should prioritize classical methods for general forecasting and deploy LLMs selectively on constraint-heavy, low-volatility tasks.

Code, data, and trained selectors: https://github.com/YuJ-Li/COMP545_Final_Project

Acknowledgments

We thank the Context-is-Key authors for making their benchmark generators publicly available, enabling reproducible research on context-aware forecasting.

Author Contributions

Kazi Ashhab Rahman: Designed experiments, implemented ARIMA, ETS and mixtral models. Implemented the XGBoost selector.

Yujin Li: Designed experiments, implemented Llama and GPT-4o-mini. Conducted data analysis.

Nusaibah Binte Rawnak: Designed experiments and the XGBoost selector. Primarily wrote the report and conducted data analysis.

Arjun Ashok (Mentor): Provided guidance on experimental design, reviewed results, and gave feedback on the manuscript.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*, fifth edition. John Wiley & Sons.

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2001. Freeway performance measurement system: mining loop detector data. *Transportation research record*, 1748(1):96–102.

Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. 2023. Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97:101819.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Juan L Gamella, Peter Bühlmann, and Jonas Peters. 2024. The causal chambers: Real physical systems as a testbed for ai methodology. *arXiv preprint arXiv:2404.11341*.

Everette S Gardner Jr. 1985. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.

Rob J Hyndman and George Athanasopoulos. 2018. *Forecasting: principles and practice*. OTexts.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation models for time series analysis: A tutorial and survey. *arXiv preprint arXiv:2403.14735*.

Kyle MacDonald et al. 2025. The memorization problem: Can we trust llms' economic forecasts? *Federal Reserve Working Paper*. Placeholder - update with arXiv number when available.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Manajit Sengupta, Yu Xie, Anthony Lopez, Aron Habte, Galen Maclaurin, and James Shelby. 2018. The national solar radiation data base (nsrdb). *Renewable and sustainable energy reviews*, 89:51–60.

U.S. Bureau of Labor Statistics. 2024. Unemployment rate [various locations]. Accessed on 2024-08-30, retrieved from FRED.

Ville de Montréal. 2020. Interventions des pompiers de montréal. Updated on 2024-09-12, accessed on 2024-09-13.

Andrew Robert Williams, Kashif Rasul, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, Sahil Garg, et al. 2024. Context is key: A benchmark for forecasting with essential textual information. *arXiv preprint arXiv:2410.18959*.

A Data Generator metadata samples

Table 5: Time Series Instances: First 5 Tasks from CashDepletedinATMScenario Domain

ID	History (first 6)	Future (first 6)	Len	Domain
task_000	4.56, 3.79, 3.67, 5.71, 4.48, 6.29 *...[162]*	9.73, 11.32, 8.88, 9.29, 7.54, 10.59 *...[50]*	224	atm_w.
task_001	26.31, 35.12, 14.84, 23.33, 16.03, 18.12 *...[106]*	24.59, 28.50, 6.99, 14.02, 15.09, 19.08 *...[50]*	168	atm_w.
task_002	9.81, 14.05, 10.62, 13.93, 17.80, 25.15 *...[106]*	17.81, 28.53, 30.91, 19.32, 16.62, 19.45 *...[50]*	168	atm_w.
task_003	6.85, 10.53, 9.63, 13.13, 13.96, 24.89 *...[106]*	15.73, 27.82, 29.49, 20.21, 17.49, 21.48 *...[50]*	168	atm_w.
task_004	18.57, 14.22, 17.76, 30.54, 37.73, 28.29 *...[162]*	17.25, 13.98, 14.76, 24.34, 36.81, 29.87 *...[50]*	224	atm_w.

Note: First 6 values shown; remaining points in brackets.

Table 6: Sample tasks from the CashDepletedinATMScenario domain.

ID	Generator	Seed	History	Horizon	Mean	Std	Trend	Vol.	Ctx Len
task_000	CashDepletedinATMScenarioTask	42	168	56	6.48	3.11	0.027	0.479	270
task_001	CashDepletedinATMScenarioTask	43	112	56	18.96	8.47	-0.045	0.447	269
task_002	CashDepletedinATMScenarioTask	44	168	56	15.68	5.32	-0.002	0.339	269
task_003	CashDepletedinATMScenarioTask	45	112	56	15.57	6.51	-0.026	0.418	269
task_004	CashDepletedinATMScenarioTask	46	168	56	20.86	8.84	-0.010	0.424	269

Note: All tasks generated from ATM withdrawal data in England. History and Horizon measured in days (timesteps). Context (269–270 characters): "This is the number of cash withdrawals from an automated teller machine (ATM) in an arbitrary location in England. Consider a scenario where the ATM runs out of cash during the forecast period."

Prompt Example

```
You are a time series forecasting expert. Context: This is the number of cash withdrawals from an ATM in England. Cash is depleted in the ATM from 1997-06-06 for 11 days, resulting in no withdrawals during that period. Historical values (last 50): 4.56, 3.79, 5.71, ..., 20.86 Task: Predict the next 56 values. Output ONLY 56 comma-separated numbers.
```

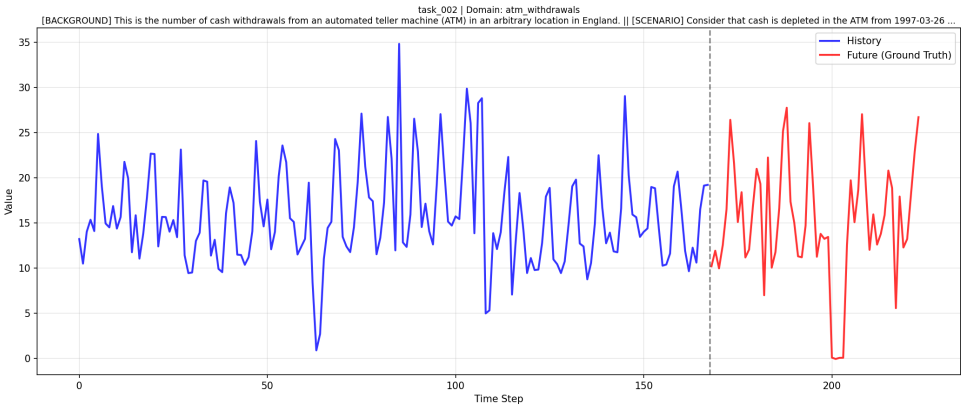

Figure 4: ARIMA Plot for task_002 in ATM Withdrawals domain.

B Per-Domain Performance Analysis

Table 7: Per-Domain Performance Analysis: Baseline, Mistral, Selector, and Oracle Policies

Domain	Baseline NMAE	Mistral NMAE	Selector NMAE	Oracle NMAE	Mistral Win %	Selector Gap
CashDepletedinATMScenario	0.423	0.596	0.423	0.423	0.0	0.000
DecreaseInTraffic	0.480	0.705	0.452	0.452	30.0	0.000
DirectNormalIrradianceFromCloudStatus	1.236	0.705	0.684	0.684	90.0	0.000
ElectricityIncreaseInPredictionTask	0.311	0.375	0.311	0.311	20.0	0.000
FullCausalContextExplicitEquationBivarLinSVAR	0.662	0.810	0.651	0.651	40.0	0.000
OracleePredUnivariateConstraints	0.765	0.593	0.392	0.300	30.0	0.093
PredictableSpikes	0.280	0.383	0.280	0.280	0.0	0.000
STLPredTrendMultiplierWithMediumDescription	0.404	0.478	0.403	0.403	20.0	0.000
SensorMaintenance	0.565	0.814	0.550	0.550	20.0	0.000
SolarPowerProduction	3.733	3.336	3.005	3.005	60.0	0.000
SpeedFromLoadTask	1.165	1.175	1.163	1.161	80.0	0.002
UnemploymentCountyUsingSingleStateData	0.173	0.189	0.169	0.164	20.0	0.005
Mean	**0.850**	**0.846**	**0.724**	**0.699**	**34.2**	**0.008**

Note: NMAE = Normalized Mean Absolute Error (lower is better). Mistral Win % = percentage of tasks where Mistral beats baseline. Selector Gap = difference between Selector and Oracle NMAE. Domains with 0% win rate show no context benefit.