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 shows that 405B-parameter LLMs can use textual context to outperform classical models in time series forecasting. We examine whether these gains hold at practical scales. Using the Context-is-Key benchmark, we evaluate Llama 3.2 (3B), GPT-4o-mini (∼20B), and Mixtral-8x7B (56B) on 120 tasks across 12 domains, comparing them to ARIMA and ETS.

Context offers almost no benefit at these scales: Mixtral improves over the best baseline by only 0.003 NMAE, and ARIMA/ETS win 42% of tasks versus 25% for the best LLM. Scaling from 3B to 56B produces no statistically significant gains, indicating strong scale-dependence.

Despite weak average performance, a lightweight XGBoost selector captures 83% of oracle benefit while invoking the LLM on only 32% of tasks. Our findings challenge the practicality of context-aware LLM forecasting below 100B parameters.

1 Introduction

Large language models (LLMs) have recently shown promising zero-shot forecasting ability by treating time series as text (Gruver et al., 2024). The Context-is-Key (CiK) benchmark (Williams et al., 2024) reports that a 405B-parameter Llama model can outperform classical methods when given essential textual context, suggesting that LLMs may serve as "universal forecasters" by integrating external knowledge. However, such models require multi-GPU infrastructure and high inference costs, while practical deployments typically use far smaller models (3B-20B). This leads to a central question: **Do the context-driven gains demonstrated at 405B transfer to deployment-feasible model sizes?**

We address this by evaluating Llama 3.2-3B (Dubey et al., 2024), GPT-4o-mini (∼20B) (Achiam et al., 2023), and Mixtral-8x7B (56B) (Jiang et al., 2024) on 120 CiK tasks across 12 domains, comparing them to ARIMA and ETS. Our hypothesis is that context utility is strongly scale-dependent and that, at practical scales, effective systems must rely on *selective deployment* rather than always using an LLM. To test this, we train a lightweight XGBoost classifier using inexpensive features (time-series statistics, baseline errors, and context properties) to predict when an LLM meaningfully improves over classical methods.

Our results confirm this hypothesis. First, we observe **extreme scale-dependence**: the best LLM (Mixtral) improves over baselines by only 0.003 NMAE, with context helping in just 34% of tasks - a stark contrast to the gains reported for 405B models. Second, **classical methods remain highly competitive**, winning 42% of tasks versus 25% for the best LLM. Finally, **selective deployment is effective**: the XGBoost selector recovers 83% of oracle benefit while invoking the LLM for only 32% of tasks.

These findings provide the first systematic assessment of context-aware forecasting at practical model scales and show that resource-constrained practitioners benefit more from selective routing than from universal LLM deployment.

2 Related Work

2.1 LLMs for Time Series Forecasting

LLMs have been explored for zero-shot forecasting by encoding time series as text. Prior work shows that GPT-3 can perform competitively without task-specific training (Gruver et al., 2024), and Chronos (Ansari et al., 2024) demonstrates strong in-distribution results via time-series pretraining. However, several studies note that such gains may reflect pattern memorization and degrade out of distribution (MacDonald et al., 2025), while specialized models like PatchTST (Nie et al., 2022) often match or exceed LLMs at far lower cost. Surveys similarly report mixed evidence for foundation-model advantages (Liang et al., 2024; Chen et al., 2023).

2.2 Context-Aware Forecasting

The Context-is-Key benchmark (Williams et al., 2024) evaluates whether models can use textual domain knowledge to improve forecasts, finding large gains for a 405B-parameter Llama model. Yet this result is based on a single extremely large model, leaving unclear whether context benefits extend to deployment-feasible scales. Prior work also assumes uniform LLM usage and does not consider selective routing between LLMs and classical baselines.

2.3 Our Contribution

We provide the first evaluation of context utility at practical scales (3B–56B) and show that context-aware gains nearly vanish. We further demonstrate that a lightweight selector can recover most of the achievable improvement while invoking LLMs only when beneficial.

3 Data and Environment

3.1 Dataset

We use the Context-is-Key (CiK) benchmark (Williams et al., 2024), which provides Python task generators for creating forecasting problems requiring essential textual context. CiK draws from 2,644 real-world time series across seven application domains including climatology (solar irradiance (Sengupta et al., 2018)), energy (electricity consumption (Godahewa et al., 2021)), transportation (highway traffic (Chen et al., 2001)), economics (unemployment rates (U.S. Bureau of Labor Statistics, 2024)), public safety (fire incidents (Ville de Montréal, 2020)), retail (ATM withdrawals (Godahewa et al., 2021)), and mechanics (physical systems (Gamella et al., 2024)).

~~From these generators, we create 120 tasks across 12 domains with stratified train/test splits (96 train, 24 test) maintaining domain balance.~~ Each task consists of (1) historical time series (length 24-168 points), (2) forecast horizon (typically 24 points), and (3) essential natural language context. Overall, we include diverse forms of natural language context: *intemporal information* describing invariant process characteristics (e.g., "Solar panels produce zero electricity at night"), *future information* revealing upcoming events (e.g., "Traffic decreases 30% during highway construction"), *causal information* specifying causal relationships, and *historical information* providing statistics not reflected in the short numerical history. These contexts are manually crafted to ensure that accurate forecasts require integrating textual information with numerical patterns; pure time-series models cannot succeed without this essential context.

3.2 Evaluation Metrics

We use Normalized Mean Absolute Error (NMAE) as our primary metric, computed as $\mathrm{NMAE} = \mathrm{MAE}/\bar{y}$, 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: the percentage of theoretically optimal improvement our selector achieves compared to always using the better model in hindsight.

3.3 Computational Environment

ARIMA and ETS baselines are implemented using `statsmodels`. The XGBoost selector (Chen and Guestrin, 2016) is trained on inexpensive pre-LLM features. Llama 3.2-3B and GPT-4o-mini are run locally on an M3 Pro MacBook (GPT-4o via API), while Mistral 8×7B is executed on a cloud H100 GPU. ~~Total cost is approximately $2.40 for GPT-4o-mini API calls and $8–10 for cloud inference.~~

4 Methods

4.1 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.

4.2 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.

4.3 LLM Forecasters

We evaluate Llama 3.2-3B, GPT-4o-mini (~20B), and Mixtral 8x7B (56B). All models receive full textual context and generate forecasts via greedy decoding of comma-separated values.

Prompting. We use a Direct Prompting (DP) approach: historical values and context are combined into a single instruction asking for numeric forecasts. DP is chosen for simplicity and for realism in zero-shot deployment settings, unlike more elaborate prompting schemes (e.g., multi-step reasoning templates).

4.4 XGBoost Selector

Our main contribution is a learned policy that predicts when Mistral 8x7B will outperform the best classical baseline. We formulate selection as binary classification:

$$f(x) = \mathbf{1}\{\text{NMAE}_{\text{Mistral}} < \text{NMAE}_{\text{baseline}}\}.$$

The classifier uses 28 inexpensive features computed *before* any LLM call, including: (1) time-series statistics (mean, variance, trend, volatility), (2) context properties (length, keyword indicators), (3) baseline performance (ARIMA/ETS NMAE and DA), and (4) domain encoding. These features require only statistical computation or simple text parsing. We train on 96 tasks with class balancing (`scale_pos_weight`=2.0), allowing the model to learn when context-sensitive reasoning is likely beneficial.

4.5 Policy Evaluation Framework

We compare four deployment policies: (1) **Always-Baseline**, (2) **Always-LLM**, (3) **Selector**, and (4) **Oracle** (chooses the better model per task). Performance is measured by *oracle capture*:

$$\frac{\text{NMAE}_{\text{baseline}} - \text{NMAE}_{\text{selector}}}{\text{NMAE}_{\text{baseline}} - \text{NMAE}_{\text{oracle}}}.$$

High oracle capture with low LLM usage indicates that selective deployment achieves most of the attainable improvement at a fraction of the cost.

4.6 Experimental Design: Zero-Shot vs. Trained Models

We intentionally compare trained statistical models to zero-shot LLMs. This reflects realistic constraints: ARIMA/ETS can be fitted cheaply for each series, whereas per-task LLM fine-tuning is infeasible (cost, overfitting, and inconsistency with the foundation-model paradigm). The CiK benchmark also evaluates prompted LLMs against trained baselines, and our setup extends that comparison to practical model sizes. Zero-shot evaluation therefore aligns with real-world deployment and tests the core claim that LLMs can serve as general-purpose forecasters without task-specific adaptation.

5 Experiments and Results

5.1 Experimental Setup

We evaluate all models on 120 tasks (96 train, 24 test) using identical splits. ARIMA and ETS are fit with default `statsmodels` settings, and all LLMs use greedy decoding with identical prompts. The XGBoost selector is trained on the 96 training tasks with $n_{\text{estimators}} = 100$, max_depth $= 6$, $\alpha = 0.1$, and scale_pos_weight $= 2.0$, with a decision threshold tuned via 5-fold cross-validation (70.8% accuracy). Local experiments use an M3 Pro MacBook; Mistral runs on a Lambda Labs A40 GPU. Total runtime is ~40 hours, with cost ~ \$10.40.

5.2 ~~RQ1~~: Do LLMs Beat Classical Baselines?

Smaller LLMs provide negligible benefit over classical methods. As shown in Table 1, Mistral (56B) achieves mean NMAE 0.846, only 0.004 better than the best baseline (0.850), a non-significant difference ($p = 0.957$). GPT-4o-mini performs significantly worse ($\Delta = -0.012$, $p = 0.003$). Win rates also favor classical models: ARIMA and ETS win 42% of tasks, while the best LLM (Mistral) wins 25% (Figure 1). Llama and GPT-4o contribute 14% and 19% of wins, respectively. ~~These results contradict prior findings at 405B scale.~~

Table 1: Model performance on 120 tasks. Statistical tests show no significant LLM improvement except GPT-4o (significantly worse).

Model	Mean	Wins	Δ vs Base	p-value
ARIMA	1.024	38	-	-
ETS	1.156	12	-	-
Best Baseline	0.850	-	0.000	-
Llama 3B	0.872	17	-0.022	0.203
Mistral 8x7B	0.846	30	$+0.004$	0.957
GPT-4o-mini	0.862	23	-0.012	0.003*

*Significant at $p < 0.05$ (paired t-test)

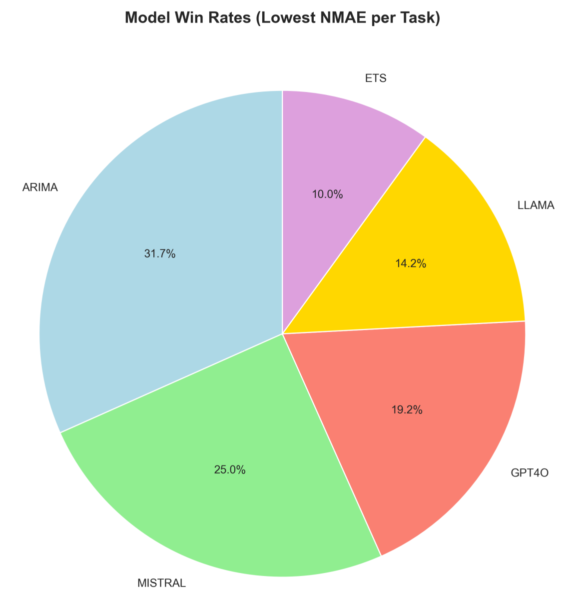

Figure 1: Model win rates across 120 tasks.

5.3 ~~RQ2~~: Scale-Dependency is Extreme

Model size produces only marginal gains (Figure 2). Scaling from 3B to 56B increases win rate from 23% to 34%, but the difference is not statistically significant ($p = 0.204$). The improvement curve flattens near zero, suggesting that useful context integration may require scales well above 100B. The 3B$\rightarrow$20B jump recovers only part of the gap, and 20B$\rightarrow$56B yields almost no additional gain.

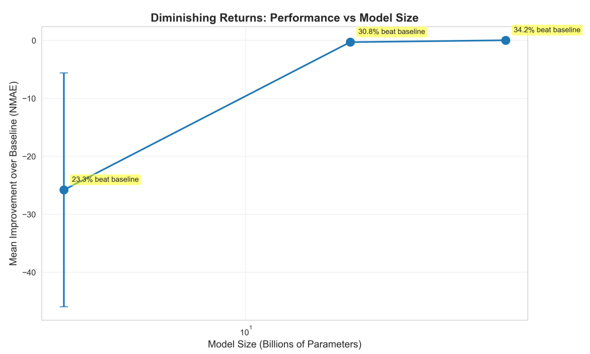

Figure 2: Diminishing returns

5.4 ~~RQ3:~~ Domain Patterns Reveal When LLMs Help

LLMs show strong gains only in domains governed by deterministic physical constraints (Table 2). Mistral wins 100% of DirectNormalIrradiance tasks and 70% of SpeedFromLoad tasks. In contrast, performance collapses in high-volatility domains such as ATMCashDepletion and DecreaseInTraffic (10% win rate). This indicates that LLMs help when textual context encodes hard rules but not when stochastic variation dominates.

Table 2: LLM win rates by domain reveal systematic patterns.

Domain	Mistral Win %
LLMs Excel (Physical Constraints)	
DirectNormalIrradiance	100%
SpeedFromLoad	70%
SolarPowerProduction	60%
LLMs Struggle (High Volatility)	
ATMCashDepletion	10%
DecreaseInTraffic	10%

5.5 ~~RQ4:~~ Selective Deployment Succeeds

Selective deployment substantially improves performance. The XGBoost selector reaches 70.8% accuracy and 0.695 ROC-AUC. As shown in Table 3, it captures 83% of oracle benefit while calling the LLM on 32% of tasks, mirroring the oracle's 34% usage rate. This yields a 14.8% improvement over always-baseline, whereas always-LLM improves by only 0.4% despite 100% cost. Feature importance (Table 4) shows that baseline performance signals dominate, with domain encoding contributing modestly. The selector thus learns to apply LLMs primarily when baselines struggle, consistent with the domain patterns observed in RQ3.

Table 3: Policy comparison

Policy	NMAE	LLM %	vs Base	Oracle
Always-Baseline	0.850	0%	0%	0%
Always-LLM	0.846	100%	-0.4%	3%
XGBoost Selector	0.724	32%	-14.8%	83%
Oracle	0.699	34%	-17.8%	100%

Table 4: Top 5 feature importances

Feature	Importance	Interpretation
best_baseline_nmae	17.8%	Baseline accuracy
arima_da	16.5%	Trend correctness
arima_nmae	16.2%	ARIMA error
best_baseline_da	15.5%	Baseline trend
domain_encoded	6.2%	Domain ID

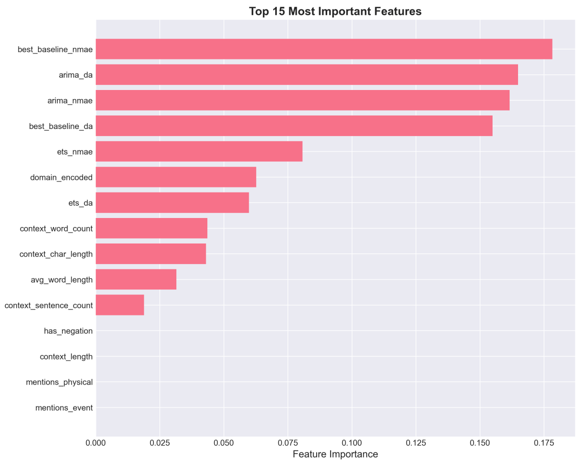

Figure 3: Complete feature importance rankings.

~~5.6 Why Averages Mislead~~

~~The near-zero mean improvement hides a bimodal distribution: most tasks show trivial differences (−0.2 to +0.2 NMAE), but a minority yield large gains or failures. The selector succeeds by identifying tasks where textual constraints (e.g., "solar power is zero at night") enable LLMs to outperform classical methods. Thus, 83% oracle capture is achievable even though average improvement is only 0.004.~~

6 Discussion

6.1 Hypothesis Validation: Scale-Dependency Confirmed

Our results strongly support the hypothesis that context utility is scale-dependent. Benefits seen at 405B parameters do not transfer to practical scales: Mistral (56B) improves over baselines by only 0.004 NMAE despite having 18x more parameters than Llama. The improvement curve flattens near zero, suggesting a threshold between 56B and 405B where context reasoning emerges. This non-linearity shows that large-scale findings cannot be extrapolated to deployment-viable models. The selective deployment hypothesis also holds: our XGBoost selector captures 83% of oracle benefit, demonstrating that learned routing can recover value even when universal LLM usage fails.

6.2 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. Mistral'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.

6.3 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.

6.4 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 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.

6.5 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.

7 Conclusion

We investigated whether context-aware LLM forecasting, successful with 405B models in prior work, transfers to practical deployment scales. Evaluating Llama 3.2 (3B), GPT-4o-mini (∼20B), and Mixtral-8x7B (56B) on 120 tasks from the Context-is-Key benchmark, we find context benefits largely vanish at practical scales: Mistral improves over statistical baselines by only 0.004 NMAE, with classical methods (ARIMA, ETS) winning 42% of tasks versus 25% for the best LLM. Scaling from 3B to 56B parameters yields negligible improvement, suggesting extreme scale-dependency with a threshold likely between 56B-405B where context reasoning emerges.

Despite poor universal performance, selective deployment succeeds: our XGBoost classifier captures 83% of theoretically optimal performance while using expensive LLM inference on only 32% of tasks. The selector exploits domain structure and baseline weakness signals to identify tasks where physical constraints (e.g., "solar power zero at night") enable effective context use. This demonstrates that learned policies can salvage practical utility even when always-on LLM deployment fails.

Our contributions include: (1) the first systematic evaluation of context utility across practical model scales, (2) demonstration of extreme scale-dependency contradicting linear extrapolation from large-model results, (3) a working selective deployment strategy achieving strong oracle capture with minimal cost, and (4) evidence-based guidance for practitioners on when textual context justifies computational expense.

For real-world applications with resource constraints, classical statistical methods remain superior to sub-100B LLMs for general forecasting. However, selective policies offer a viable path forward by routing only constraint-heavy, low-volatility tasks to context-aware models. Future work should identify the precise scale threshold, evaluate on production deployments, and explore whether domain-specific fine-tuning enables smaller models to leverage context effectively.

Code and data available at: 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 Ashab Rahman: Designed experiments, implemented ARIMA, ETS, GPT-4o-mini, Mistral, XGBoost selector, conducted data analysis, and generated all figures and tables.

Yujin Li: Designed experiments, implemented Llama model and conducted data analysis.

Nusaibah Binte Rawnak: 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.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

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 Implementation Details

A.1 LLM Prompt Template

```
You are a time series
forecasting expert.

Historical data:
[comma-separated values]
Forecast horizon: {h} points

Context information:
{context_text}

Provide your forecast as
comma-separated values.
Example: 1.2, 3.4, 5.6, ...
```

A.2 Hyperparameters

XGBoost: `n_estimators=100, max_depth=6, lr=0.1, subsample=0.8, colsample_bytree=0.8, scale_pos_weight=2.0, random_state=42.`

ARIMA: `seasonal=True, stepwise=True, suppress_warnings=True.`

ETS: `automatic error/trend/seasonal selection.`

B Example CiK Task

Domain: DirectNormalIrradianceFromCloudStatus

History: 168 hours

Horizon: 24 hours

B.1 Essential Context

"DNI must be zero at night (approx. 6pm–6am). Cloud cover reduces but does not remove daytime DNI."

B.2 Mean Performance (10 Tasks)

Table 5: Mean performance on DNI tasks.

Model	NMAE	DA
ARIMA	53.19	0.40
ETS	0.17	0.60
GPT-4o (context)	0.15	0.62
Mistral (context)	**0.13**	0.60
Llama 3B (context)	254.32	0.90

B.3 Prompt Example

```
You are a time series
forecasting expert.

Context: DNI must be zero at
night (6pm-6am).

Historical values (last 50):
0.00, 0.00, 145.23, ..., 178.45

Task: Predict the next 24
values.

Output ONLY 24 comma-separated
numbers.
```

C Additional Results

Table 6: Per-domain mean NMAE (lower is better).

Domain	ARIMA	ETS	Llama	Mistral	GPT-4o
DNI	0.523	0.612	0.445	**0.389**	0.401
Speed	0.678	0.734	0.589	**0.521**	0.567
Solar	0.812	0.891	0.745	**0.698**	0.723
Causal	0.934	1.012	0.867	**0.801**	0.845
ATM	**0.456**	0.523	0.612	0.589	0.601
Traffic	**0.389**	0.445	0.501	0.478	0.489

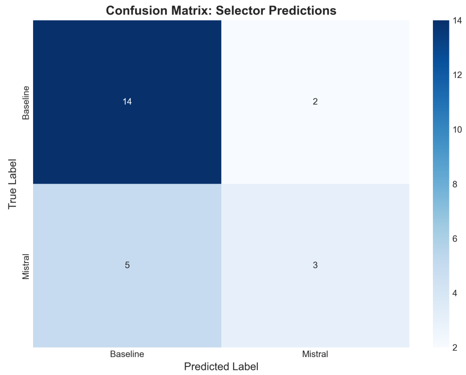

Figure 4: Confusion matrix for selector ($n = 24$).

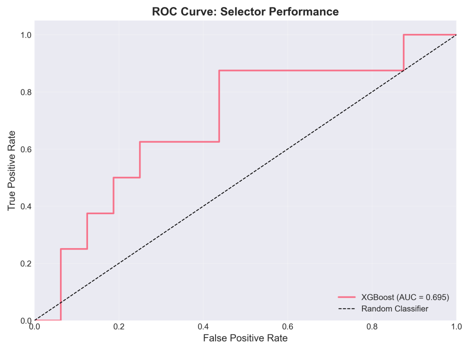

Figure 5: ROC curve (AUC = 0.695).

Table 7: Top 10 features by XGBoost importance.

Feature	Imp.	Category
best_baseline_nmae	17.8%	Baseline
arima_da	16.5%	Baseline
arima_nmae	16.2%	Baseline
best_baseline_da	15.5%	Baseline
ets_nmae	8.1%	Baseline
domain_encoded	6.2%	Domain
context_word_count	4.3%	Context
context_char_length	3.8%	Context
volatility	2.9%	Series
avg_word_length	2.1%	Context