

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 (~ 20 B), 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 (~ 20 B) (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

80 similarly report mixed evidence for foundation-
81 model advantages (Liang et al., 2024; Chen et al.,
82 2023).

83 2.2 Context-Aware Forecasting

84 The Context-is-Key benchmark (Williams et al.,
85 2024) evaluates whether models can use textual
86 domain knowledge to improve forecasts, finding
87 large gains for a 405B-parameter Llama model.
88 Yet this result is based on a single extremely large
89 model, leaving unclear whether context benefits
90 extend to deployment-feasible scales. Prior work
91 also assumes uniform LLM usage and does not con-
92 sider selective routing between LLMs and classical
93 baselines.

94 2.3 Our Contribution

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

101 3 Data and Environment

102 3.1 Dataset

103 We use the Context-is-Key (CiK) bench-
104 mark (Williams et al., 2024), which provides
105 Python task generators for creating forecasting
106 problems requiring essential textual context. CiK
107 draws from 2,644 real-world time series across
108 seven application domains including climatology
109 (solar irradiance (Sengupta et al., 2018)), energy
110 (electricity consumption (Godahewa et al., 2021)),
111 transportation (highway traffic (Chen et al.,
112 2001)), economics (unemployment rates (U.S.
113 Bureau of Labor Statistics, 2024)), public safety
114 (fire incidents (Ville de Montréal, 2020)), retail
115 (ATM withdrawals (Godahewa et al., 2021)), and
116 mechanics (physical systems (Gamella et al.,
117 2024)).

118 From these generators, we create 120 tasks
119 across 12 domains with stratified train/test splits
120 (96 train, 24 test) maintaining domain balance.
121 Each task consists of (1) historical time series
122 (length 24–168 points), (2) forecast horizon (typi-
123 cally 24 points), and (3) essential natural language
124 context. Overall, we include diverse forms of nat-
125 ural language context: *intemporal information* de-
126 scribing invariant process characteristics (e.g., "So-
127 lar panels produce zero electricity at night"), *future*

information revealing upcoming events (e.g., "Traf-
128 fic decreases 30% during highway construction"),
causal information specifying causal relationships,
130 and historical information providing statistics not
reflected in the short numerical history. These con-
132 texts are manually crafted to ensure that accurate
133 forecasts require integrating textual information
134 with numerical patterns; pure time-series models
135 cannot succeed without this essential context. 136

137 3.2 Evaluation Metrics

We use Normalized Mean Absolute Error (NMAE)
138 as our primary metric, computed as $NMAE =$
139 MAE/\bar{y} , where \bar{y} is the mean of historical values.
140 NMAE enables fair comparison across domains
141 with vastly different scales (e.g., solar irradiance
142 in W/m^2 versus ATM withdrawals in dollars). We
143 also report Directional Accuracy (DA), the fraction
144 of forecasts correctly predicting upward or down-
145 ward trends relative to the last historical value, to
146 evaluate qualitative forecast quality. For policy
147 evaluation, we measure oracle capture: the per-
148 centage of theoretically optimal improvement our
149 selector achieves compared to always using the
150 better model in hindsight. 151

152 3.3 Computational Environment

ARIMA and ETS baselines are implemented using
153 statsmodels. The XGBoost selector (Chen and
154 Guestrin, 2016) is trained on inexpensive pre-LLM
155 features. Llama 3.2-3B and GPT-4o-mini are run
156 locally on an M3 Pro MacBook (GPT-4o via API),
157 while Mistral 8×7B is executed on a cloud H100
158 GPU. Total cost is approximately \$2.40 for GPT-
159 4o-mini API calls and \$8-10 for cloud inference. 160

161 4 Methods

162 4.1 Research Hypothesis

We hypothesize that **context utility in LLM fore-
163 casting is scale-dependent**: gains observed with a
164 405B model do not transfer to practical scales (3B–
165 56B). We therefore expect (1) small and mid-sized
166 LLMs to show limited improvement over statistical
167 baselines, (2) minimal performance gains from pa-
168 rameter scaling within this range, and (3) selective
169 deployment to recover most of the achievable bene-
170 fit by identifying tasks where context meaningfully
171 helps. 172

173 4.2 Baseline Models

We compare LLMs against two standard fore-
174 casting methods. **AutoARIMA** (Box et al., 175

176 2015; Hyndman and Athanasopoulos, 2018) selects
 177 ARIMA(p, d, q) parameters via stepwise search,
 178 while ETS (Gardner Jr, 1985) models error, trend,
 179 and seasonality with exponential smoothing. For
 180 each task, the *best baseline* is

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

181 where NMAE normalizes MAE by the mean of his-
 182 torical observations, enabling cross-domain com-
 183 parability.

184 4.3 LLM Forecasters

185 We evaluate Llama 3.2-3B, GPT-4o-mini ($\sim 20\text{B}$),
 186 and Mixtral 8x7B (56B). All models receive full
 187 textual context and generate forecasts via greedy
 188 decoding of comma-separated values.

189 **Prompting.** We use a Direct Prompting (DP) ap-
 190 proach: historical values and context are combined
 191 into a single instruction asking for numeric fore-
 192 casts. DP is chosen for simplicity and for realism in
 193 zero-shot deployment settings, unlike more elabo-
 194 rate prompting schemes (e.g., multi-step reasoning
 195 templates).

196 4.4 XGBoost Selector

197 Our main contribution is a learned policy that pre-
 198 dicted when Mistral 8x7B will outperform the best
 199 classical baseline. We formulate selection as binary
 200 classification:

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

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

212 4.5 Policy Evaluation Framework

213 We compare four deployment policies: (1) **Always-**
214 Baseline, (2) **Always-LLM**, (3) **Selector**, and (4)
 215 **Oracle** (chooses the better model per task). Perfor-
 216 mance 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
 217 that selective deployment achieves most of the at-
 218tainable improvement at a fraction of the cost. 219

220 4.6 Experimental Design: Zero-Shot vs. 221 Trained Models

We intentionally compare trained statistical mod-
 222 els to zero-shot LLMs. This reflects realistic con-
 223 straints: ARIMA/ETS can be fitted cheaply for
 224 each series, whereas per-task LLM fine-tuning
 225 is infeasible (cost, overfitting, and inconsistency
 226 with the foundation-model paradigm). The CiK
 227 benchmark also evaluates prompted LLMs against
 228 trained baselines, and our setup extends that com-
 229 parison to practical model sizes. Zero-shot evalua-
 230 tion therefore aligns with real-world deployment
 231 and tests the core claim that LLMs can serve as
 232 general-purpose forecasters without task-specific
 233 adaptation. 234

235 5 Experiments and Results

236 5.1 Experimental Setup

We evaluate all models on 120 tasks (96 train, 24
 237 test) using identical splits. ARIMA and ETS are
 238 fit with default `statsmodels` settings, and all
 239 LLMs use greedy decoding with identical prompts.
 240 The XGBoost selector is trained on the 96 training
 241 tasks with $n_{\text{estimators}} = 100$, $\text{max_depth} = 6$,
 242 $\alpha = 0.1$, and $\text{scale_pos_weight} = 2.0$, with a de-
 243 cision threshold tuned via 5-fold cross-validation
 244 (70.8% accuracy). Local experiments use an M3
 245 Pro MacBook; Mistral runs on a Lambda Labs
 246 A40 GPU. Total runtime is ~ 40 hours, with cost
 247 $\sim \$10.40$. 248

249 5.2 RQ1: Do LLMs Beat Classical Baselines?

Smaller LLMs provide negligible benefit over clas-
 250 sical methods. As shown in Table 1, Mistral (56B)
 251 achieves mean NMAE 0.846, only 0.004 better
 252 than the best baseline (0.850), a non-significant
 253 difference ($p = 0.957$). GPT-4o-mini performs sig-
 254 nificantly worse ($\Delta = -0.012$, $p = 0.003$). Win
 255 rates also favor classical models: ARIMA and ETS
 256 win 42% of tasks, while the best LLM (Mistral)
 257 wins 25% (Figure 1). Llama and GPT-4o contribute
 258 14% and 19% of wins, respectively. These results
 259 contradict prior findings at 405B scale. 260

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)

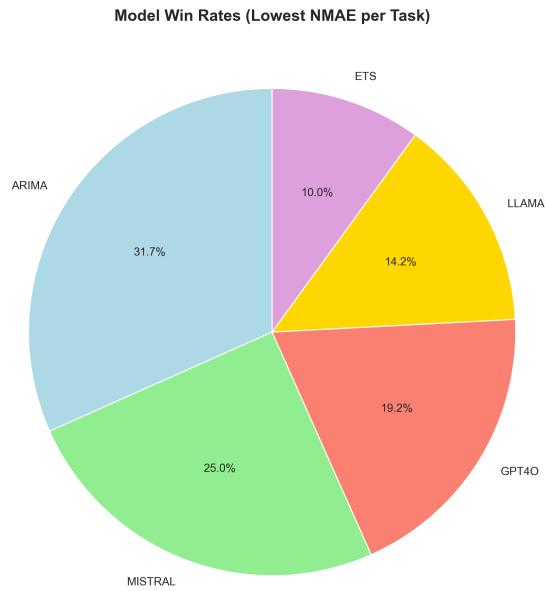


Figure 1: Model win rates across 120 tasks.

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→20B jump recovers only part of the gap, and 20B→56B yields almost no additional gain.

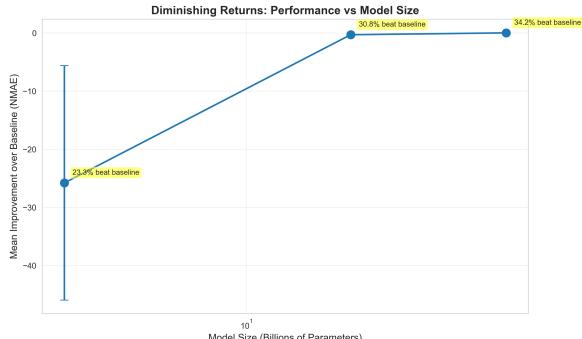


Figure 2: Diminishing returns

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 %
<i>LLMs Excel (Physical Constraints)</i>	
DirectNormalIrradiance	100%
SpeedFromLoad	70%
SolarPowerProduction	60%
<i>LLMs Struggle (High Volatility)</i>	
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

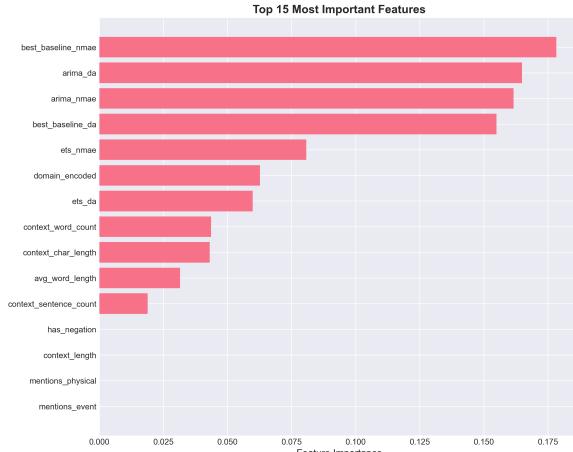


Figure 3: Complete feature importance rankings.

295 5.6 Why Averages Mislead

296 The near-zero mean improvement hides a bimodal
 297 distribution: most tasks show trivial differences
 298 (-0.2 to $+0.2$ NMAE), but a minority yield large
 299 gains or failures. The selector succeeds by identi-
 300 fying tasks where textual constraints (e.g., “solar
 301 power is zero at night”) enable LLMs to outper-
 302 form classical methods. Thus, 83% oracle capture
 303 is achievable even though average improvement is
 304 only 0.004.

305 6 Discussion

306 6.1 Hypothesis Validation: Scale-Dependency 307 Confirmed

308 Our results strongly support the hypothesis that
 309 context utility is scale-dependent. Benefits seen
 310 at 405B parameters do not transfer to practical
 311 scales: Mistral (56B) improves over baselines by
 312 only 0.004 NMAE despite having 18x more param-
 313 eters than Llama. The improvement curve flattens
 314 near zero, suggesting a threshold between 56B and
 315 405B where context reasoning emerges. This non-
 316 linearity shows that large-scale findings cannot be
 317 extrapolated to deployment-viable models. The
 318 selective deployment hypothesis also holds: our
 319 XGBoost selector captures 83% of oracle bene-
 320 fit, demonstrating that learned routing can recover
 321 value even when universal LLM usage fails.

322 6.2 Why Small LLMs Fail at Context 323 Integration

324 Three factors explain why smaller models strug-
 325 gle. First, **capacity limits**: they lack the reason-
 326 ing depth to jointly process numerical patterns and
 327 contextual constraints. Mistral’s small gains over
 328 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.

339 6.3 Unexpected Findings

Two results were unexpected. First, classical meth-
 340 ods were stronger than anticipated: ARIMA and
 341 ETS win 42% of tasks versus 25% for the best
 342 LLM, indicating that long-standing statistical mod-
 343 els remain competitive. Second, GPT-4o-mini’s
 344 significant degradation ($p = 0.003$) suggests that
 345 smaller API-tuned models may overreact to context-
 346 ual cues, indicating a need for more robust instruc-
 347 tion tuning.

348 6.4 Limitations and Future Work

Our study has four limitations. (1) **Model cov-
 350 erage**: we evaluate only 3B, 20B, and 56B mod-
 351 els, leaving gaps at intermediate scales (7B, 13B,
 352 70B). (2) **Dataset scope**: CiK’s 120 generated tasks
 353 may not capture real-world noise or incom-
 354 plete context. (3) **Prompting strategy**: we use Direct
 355 Prompting for deployment realism, though more
 356 complex prompting (chain-of-thought, few-shot)
 357 might improve performance. (4) **Selector simplic-
 358 ity**: hand-crafted features work well, but learned or
 359 embedding-based features could further close the
 360 oracle gap.

Future work should study intermediate scales
 362 (70B-200B), test real-world noisy context, and ex-
 363 plore domain-specific fine-tuning to improve con-
 364 text integration in smaller models.

366 6.5 Practical Implications

For practitioners using 3B-20B models, **classical
 367 statistical methods remain superior**. The gains
 368 observed at 405B parameters require scales far be-
 369 yond feasible deployment. Selective routing offers
 370 a practical alternative: by sending only appropriate
 371 tasks to LLMs, such as those involving determinis-
 372 tic constraints or explicit causal rules, meaningful
 373 improvements can be achieved at low cost. We
 374 recommend: (1) **test models at deployment scale**,
 375 (2) **start with classical baselines**, (3) **use selec-
 376 tive policies** like our XGBoost selector, and (4)

378 **prioritize constraint-heavy domains.** LLMs may
379 offer value in low-frequency, high-stakes settings,
380 whereas volatile or high-frequency tasks should
381 default to statistical methods.

382 7 Conclusion

383 We investigated whether context-aware LLM fore-
384 casting, successful with 405B models in prior work,
385 transfers to practical deployment scales. Evalu-
386 ating Llama 3.2 (3B), GPT-4o-mini (\sim 20B), and
387 Mixtral-8x7B (56B) on 120 tasks from the Context-
388 is-Key benchmark, we find context benefits largely
389 vanish at practical scales: Mistral improves over
390 statistical baselines by only 0.004 NMAE, with
391 classical methods (ARIMA, ETS) winning 42% of
392 tasks versus 25% for the best LLM. Scaling from
393 3B to 56B parameters yields negligible improve-
394 ment, suggesting extreme scale-dependency with a
395 threshold likely between 56B-405B where context
396 reasoning emerges.

397 Despite poor universal performance, selective
398 deployment succeeds: our XGBoost classifier cap-
399 tures 83% of theoretically optimal performance
400 while using expensive LLM inference on only 32%
401 of tasks. The selector exploits domain structure
402 and baseline weakness signals to identify tasks
403 where physical constraints (e.g., "solar power zero
404 at night") enable effective context use. This demon-
405 strates that learned policies can salvage practi-
406 cal utility even when always-on LLM deployment
407 fails.

408 Our contributions include: (1) the first system-
409 atic evaluation of context utility across practical
410 model scales, (2) demonstration of extreme scale-
411 dependency contradicting linear extrapolation from
412 large-model results, (3) a working selective deploy-
413 ment strategy achieving strong oracle capture with
414 minimal cost, and (4) evidence-based guidance for
415 practitioners on when textual context justifies com-
416 putational expense.

417 For real-world applications with resource con-
418 straints, classical statistical methods remain su-
419 perior to sub-100B LLMs for general forecast-
420 ing. However, selective policies offer a viable
421 path forward by routing only constraint-heavy,
422 low-volatility tasks to context-aware models. Fu-
423 ture work should identify the precise scale thresh-
424 old, evaluate on production deployments, and ex-
425 plore whether domain-specific fine-tuning enables
426 smaller models to leverage context effectively.

427 Code and data available at: https://github.com/YuJ-Li/COMP545_Final_Project

//github.com/YuJ-Li/COMP545_
Final_Project

428
429

430 Acknowledgments

We thank the Context-is-Key authors for making
431 their benchmark generators publicly available, en-
432 abling reproducible research on context-aware fore-
433 casting.
434

435 Author Contributions

Kazi Ashab Rahman: Designed experiments, im-
436 plemented ARIMA, ETS, GPT-4o-mini, Mistral,
437 XGBoost selector, conducted data analysis, and
438 generated all figures and tables.
439

Yujin Li: Designed experiments, implemented
440 Llama model and conducted data analysis.
441

Nusaiyah Binte Rawnak: Wrote the report and
442 conducted data analysis.
443

Arjun Ashok (Mentor): Provided guidance on
444 experimental design, reviewed results, and gave
445 feedback on the manuscript.
446

447 References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama
448 Ahmad, Ilge Akkaya, Florencia Leoni Aleman,
449 Diogo Almeida, Janko Altenschmidt, Sam Altman,
450 Shyamal Anadkat, et al. 2023. Gpt-4 technical report.
451 *arXiv preprint arXiv:2303.08774*.
452

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen,
453 Xiyuan Zhang, Pedro Mercado, Huibin Shen, Olek-
454 sandr Shchur, Syama Sundar Rangapuram, Sebas-
455 tian Pineda Arango, Shubham Kapoor, et al. 2024.
456 Chronos: Learning the language of time series. *arXiv*
457 *preprint arXiv:2403.07815*.
458

George EP Box, Gwilym M Jenkins, Gregory C Rein-
459 sel, and Greta M Ljung. 2015. *Time series analysis:
460 forecasting and control*, fifth edition. John Wiley &
461 Sons.
462

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin
463 Varaiya, and Zhanfeng Jia. 2001. Freeway perfor-
464 mance measurement system: mining loop detector
465 data. *Transportation research record*, 1748(1):96–
466 102.
467

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A
468 scalable tree boosting system. In *Proceedings of
469 the 22nd ACM SIGKDD International Conference
470 on Knowledge Discovery and Data Mining*, pages
471 785–794.
472

Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang,
473 and Chongshou Li. 2023. Long sequence time-series
474 forecasting with deep learning: A survey. *Informa-
475 tion Fusion*, 97:101819.
476

- 477 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey,
478 Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman,
479 Akhil Mathur, Alan Schelten, Amy Yang, Angela
480 Fan, et al. 2024. The llama 3 herd of models. *arXiv*
481 preprint arXiv:2407.21783. 529
- 482 Juan L Gamella, Peter Bühlmann, and Jonas Peters.
483 2024. The causal chambers: Real physical systems
484 as a testbed for ai methodology. *arXiv preprint*
485 arXiv:2404.11341. 530
- 486 Everette S Gardner Jr. 1985. Exponential smoothing:
487 The state of the art. *Journal of Forecasting*, 4(1):1–
488 28. 531
- 489 Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I
490 Webb, Rob J Hyndman, and Pablo Montero-Manso.
491 2021. Monash time series forecasting archive. *arXiv*
492 preprint arXiv:2105.06643. 532
- 493 Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G
494 Wilson. 2024. Large language models are zero-shot
495 time series forecasters. *Advances in Neural Informa-*
496 *tion Processing Systems*, 36. 533
- 497 Rob J Hyndman and George Athanasopoulos. 2018.
498 *Forecasting: principles and practice*. ÖTexts.
- 499 Albert Q Jiang, Alexandre Sablayrolles, Antoine
500 Roux, Arthur Mensch, Blanche Savary, Chris Bam-
501 ford, Devendra Singh Chaplot, Diego de las Casas,
502 Emma Bou Hanna, Florian Bressand, et al. 2024.
503 Mixtral of experts.
- 504 Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang,
505 Ming Jin, Dongjin Song, Shirui Pan, and Qing-
506 song Wen. 2024. Foundation models for time se-
507 ries analysis: A tutorial and survey. *arXiv preprint*
508 arXiv:2403.14735.
- 509 Kyle MacDonald et al. 2025. The memorization prob-
510 lem: Can we trust llms' economic forecasts? *Federal*
511 *Reserve Working Paper*. Placeholder - update with
512 arXiv number when available.
- 513 Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and
514 Jayant Kalagnanam. 2022. A time series is worth
515 64 words: Long-term forecasting with transformers.
516 *arXiv preprint arXiv:2211.14730*.
- 517 Manajit Sengupta, Yu Xie, Anthony Lopez, Aron Habte,
518 Galen MacLaurin, and James Shelby. 2018. The na-
519 tional solar radiation data base (nsrdb). *Renewable*
520 *and sustainable energy reviews*, 89:51–60.
- 521 U.S. Bureau of Labor Statistics. 2024. **Unemployment**
522 **rate [various locations]**. Accessed on 2024-08-30,
523 retrieved from FRED.
- 524 Ville de Montréal. 2020. **Interventions des pompiers**
525 **de montréal**. Updated on 2024-09-12, accessed on
526 2024-09-13.
- 527 Andrew Robert Williams, Kashif Rasul, Arian Kho-
528 rasani, George Adamopoulos, Rishika Bhagwatkar,
- Marin Biloš, Hena Ghonia, Nadhir Vincent Has-
sen, Anderson Schneider, Sahil Garg, et al. 2024.
Context is key: A benchmark for forecasting
with essential textual information. *arXiv preprint*
arXiv:2410.18959. 529

535 **A.1 LLM Prompt Template**

536 You are a time series
 537 forecasting expert.
 538 Historical data:
 539 [comma-separated values]
 540 Forecast horizon: {h} points
 541 Context information:
 542 {context_text}
 543 Provide your forecast as
 544 comma-separated values.
 545 Example: 1.2, 3.4, 5.6, ...

546 **A.2 Hyperparameters**

547 **XGBoost:** n_estimators=100,
 548 max_depth=6, lr=0.1, subsample=0.8,
 549 colsample_bytree=0.8,
 550 scale_pos_weight=2.0, random_state=42.
 551 **ARIMA:** seasonal=True, stepwise=True,
 552 suppress_warnings=True.
 553 **ETS:** automatic error/trend/seasonal
 554 selection.

555 **B Example CiK Task**

556 **Domain:** DirectNormalIrradianceFromCloudStatus

557 **History:** 168 hours

558 **Horizon:** 24 hours

559 **B.1 Essential Context**

560 “DNI must be zero at night (approx. 6pm–6am).
 561 Cloud cover reduces but does not remove daytime
 562 DNI.”

563 **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

564 **B.3 Prompt Example**

565 You are a time series
 566 forecasting expert.
 567 Context: DNI must be zero at
 568 night (6pm–6am).
 569 Historical values (last 50):
 570 0.00, 0.00, 145.23, ..., 178.45
 571 Task: Predict the next 24
 572 values.
 573 Output ONLY 24 comma-separated
 574 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

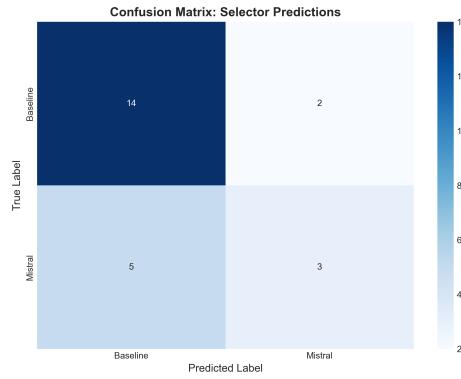


Figure 4: Confusion matrix for selector ($n = 24$).

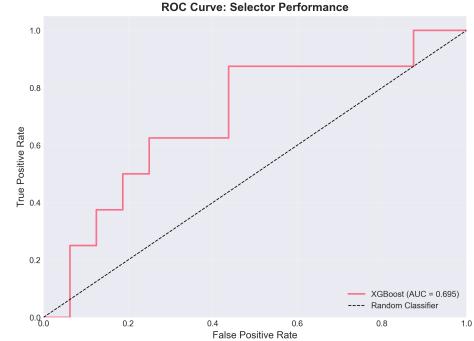


Figure 5: ROC curve (AUC = 0.695).

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