

On Identifying Why and When Foundation Models Perform Well on Time-Series Forecasting Using Automated Explanations and Rating

Michael "Xander" Widener, Kausik Lakkaraju, John Aydin, Bipav Srivastava

Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA

Abstract

Time-series forecasting models (TSFM) have evolved from classical statistical methods to sophisticated foundation models, yet understanding why and when these models succeed or fail remains challenging. Despite this known limitation, time series forecasting models are increasingly used to generate information that informs real-world actions with equally real consequences. Understanding the complexity, performance variability, and opaque nature of these models then becomes a valuable endeavor to combat serious concerns about how users should interact with and rely on these models' outputs. This work addresses these concerns by combining traditional explainable AI (XAI) methods with Rating Driven Explanations (RDE) to assess TSFM performance and interpretability across diverse domains and use cases. We evaluate four distinct model architectures: ARIMA, Gradient Boosting, Chronos (time-series specific foundation model), Llama (general-purpose; both fine-tuned and base models) on four heterogeneous datasets spanning finance, energy, transportation, and automotive sales domains. In doing so, we demonstrate that feature-engineered models (e.g., Gradient Boosting) consistently outperform foundation models (e.g., Chronos) in volatile or sparse domains (e.g., power, car parts) while providing more interpretable explanations, whereas foundation models excel only in stable or trend-driven contexts (e.g., finance).

1 Introduction

Time-series forecasting underpins critical decisions in domains such as finance, energy management, and logistics, yet model performance can vary widely across domains. Classical statistical methods like ARIMA (Shumway and Stoffer 2017) are interpretable, but often underperform on volatile or sparse data (Minarso, Adji, and Setiawan 2022). Modern approaches, ranging from feature-engineered gradient boosting (Agapitos, Brabazon, and O'Neill 2017) to foundation models like Chronos (Ansari et al. 2024), deliver high predictive accuracy but introduce opacity, raising concerns about trust and reliability. Prior work (Elsayed et al. 2021) shows the feature engineered gradient boosting can outperform all other evaluated models, including naive gradient boosting and ARIMA (an observation we also confirm in our experiments).

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

EXplainable AI (XAI) methods aim to clarify model behaviour by identifying how input features contribute to predictions (Arrieta et al. 2020). Techniques such as SHAP (Lundberg and Lee 2017) and LIME (Ribeiro, Singh, and Guestrin 2016) attribute importance at the individual prediction level (local explanations), while global approaches such as global SHAP and TsSHAP (Raykar et al. 2023) provide feature attribution across the dataset. These methods are effective for showing why a model made a given prediction, but they do not indicate whether the model behaves fairly across subgroups (Alikhademi et al. 2021) or remains reliable under different input conditions. To address this limitation, we use Rating-Driven Explanations (RDE) (Lakkaraju, Valluru, and Srivastava 2025), a causally grounded approach derived from the rating method (Lakkaraju, Srivastava, and Valtorta 2024; Lakkaraju et al. 2025). RDE measures how model outcomes change under interventions, quantifies robustness, which includes models' sensitivity to confounders and protected attributes, and compares results against biased and random baselines. This approach provides explanation from a model stability and fairness from a fairness perspective.

In this work, we use SHAP (Lundberg and Lee 2017) and LIME (Ribeiro, Singh, and Guestrin 2016) for providing local explanations, global SHAP and TsSHAP (Raykar et al. 2023) for global level feature attributions, and RDE for explanations about robustness of the models. Specifically, we address three key research questions:

RQ1: When forecasting models succeed or fail?

RQ2: What do common XAI methods tell us about why forecasting models succeed or fail?

RQ3: How can rating augment our understanding of when and why forecasting models succeed or fail?

To explore these questions, we conduct experiments on four benchmark datasets (see Table 1), which vary in frequency, volatility, and periodicity. We compare a range of forecasting models spanning classical, machine learning, and foundation-model approaches as seen in Table 3.

Our results reveal that feature-engineered models outperform foundation models in volatile domains (e.g., energy) while offering better interpretability, whereas foundation models excel in stable, trend-driven contexts (e.g., finance). These findings provide actionable guidance for selecting forecasting approaches based on domain requirements.

Dataset	Domain	Number of Series	Min Length	Max Length
Finance	Finance	6	250	250
Power	Energy	1	20,915	20,915
Pedestrian	Mobility	66	576	96,424
Car Parts	Sales	2,674	51	51

Table 1: We selected the pedestrian and Car Parts datasets from Monash (Godahewa et al. 2021), power data from collected by Tantiv4 sensors (Muppasani et al. 2023), and Yahoo Finance as representative open time-series datasets spanning diverse domains. Other widely used benchmarks (e.g., M4, ETT, WeatherBench) were excluded to maintain tractability, though inclusion in future work would further strengthen generalizability claims.

Model	Category	Params	Pretrained?	Needs FE?
ARIMA	Statistical	–	No	No
Gradient Boosting	ML (Tabular)	~100k–300k	No	Yes
Chronos	Foundation Model	47.7M	Yes	No
LLaMA (Finetuned)	LLM-based	~7B+	Yes	No

Table 2: Models used. parameters size → "Params" and reflects typical scale; feature engineering → "FE".

2 Related Work

In this section, we situate our work within relevant prior work.

2.1 Time-Series Forecasting Models (TSFM)

Foundation models (FMs), including large language models (LLMs), have recently been adapted for time-series forecasting (Zhou et al. 2023; Jin et al. 2023; Pan et al. 2024). Fine-tuned language-pretrained transformers (Zhou et al. 2023; Jin et al. 2023) and parameter-efficient adaptations (Gruver et al. 2023; Cao et al. 2023; Ekambaram et al. 2024) show competitive performance across diverse tasks. Chronos (Ansari et al. 2024) and Unified-TS (Woo et al. 2024) improve accuracy and generalization, while (Rasul et al. 2023; Das et al. 2023) explore new tokenization schemes. In explainable forecasting, (Yu et al. 2023) integrates historical prices, metadata, and news with LLMs to produce predictions and rationales. Lightweight designs (Garza and Mergenthaler-Canseco 2023; Ekambaram et al. 2024) enable real-time use, and multi-pattern integration (Talukder, Yue, and Gkioxari 2024) improves precision. Models trained from scratch, such as Lag-Llama (Gruver et al. 2023), achieve strong zero-shot performance, with further gains from temporal decomposition (Cao et al. 2023) and hybrid encoder-decoder designs (Goswami et al. 2024).

Fine-tuning remains important even with retrieval-based methods. RAF (Tire et al. 2024) and FinSrag (Xiao et al. 2025) both report higher accuracy from domain-specific adaptation. Chronos (Ansari et al. 2024) discretizes time-series through scaling and quantization, while Chronos-Bolt uses a patch-based representation and decoder outputs to improve both forecasting accuracy and inference speed. In this work, we employ Chronos-Bolt alongside a widely used general-purpose foundation model, LLaMA, and its finetuned variants as TSFM.

2.2 Explainability in AI

Explainable AI (XAI) refers to processes and methods that help describe how AI models work (Arrieta et al. 2020). Feature importance methods such as SHapley Additive exPlanations (SHAP) and counterfactual explanations are the most widely-used explanation methods. The former is sensitive to feature correlations and the latter only explains a single outcome instead of the entire model (Alfeo et al. 2023). Users benefit more from integrated local and global explanations, yet the majority of XAI methods have focused on the former (Hoffman et al. 2023). We use traditional XAI methods, to explain predictions, both at the local and global level, but these do not say much about a model’s overall fairness or robustness (Alikhademi et al. 2021). For that, we turn to RDE, which is causally grounded and better suited to communicate how the model behaves across different scenarios on a global level.

2.3 Explainability in Time-Series Forecasting Models (TSFMs)

The most common methods in time-series forecasting (Arsenault, Wang, and Patenaude 2025) are Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016) & SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017). SHAP enables users to get a sense of interpretable models’ feature attribution with both magnitude and direction. Similar to SHAP (Lundberg and Lee 2017), LIME (Ribeiro, Singh, and Guestrin 2016) calculates feature attributions using a local linear model to estimate the importance of attributes in a task. When adapted to time-series forecasting, it can be used to examine feature importance within a model or gain insights into the influence of different segments of a time-series on the forecast (Schlegel et al. 2021).

Traditional XAI methods for TSFMs rely on feature attribution and visualization methods. Model-agnostic methods such as SHAP are widely used (Mokhtari, Higdon, and Başar 2019; Schlegel et al. 2019; Zhang et al. 2022), along with partial dependence plots (Greenwell 2017) to estimate marginal contributions of input variables and forecast outputs. These methods have been adapted to consider time-dependent inputs, but they are often limited in capturing causality or robustness under input perturbation (Lakkaraju et al. 2025). In deep learning-based TSFMs, attention mechanisms have also been used to show influential time steps or variables in multivariate forecasting tasks (Abbasimehr and Paki 2022; Li et al. 2019; Shih, Sun, and Lee 2019). However, such mechanisms are typically embedded within the models and therefore may not be reliable indicators to measure causal influence. Though these methods have been useful for prediction interpretability, they tend to disproportionately focus on either local or global explanations. SHAP often provides local insights into single predictions. PDPs, for instance, offer aggregated trends across a dataset. In TSFMs, it is often necessary to combine both perspectives, i.e., understanding how individual time points contribute to certain forecasts (local), and how these patterns also generalize across entire data (global). Existing approaches

do not always consistently support both forms of explanation in an integrated or interactive manner, especially under stakeholder-specific constraints. In recent years, there has been a growing shift in the literature toward methods that not only explain model predictions but also empower stakeholders at different levels to interrogate model behavior through hypothesis testing and counterfactual reasoning (Çelik, İcan, and Bulut 2023; Tasin et al. 2023).

While these methods help magnify the explainability of already interpretable models (GB) some significant concessions are often required to explain more complex black box models (segment-based LIME, TsSHAP, ...). Adapting LIME to time-series forecasting is a non-trivial task. (Sivill and Flach 2022) propose an adaptation for classification, implementing LIME for time-series with a novel segmentation algorithm, a perturbation technique which aims to mimic realistic background noise, and distance algorithm specifically designed to compare time-series. (Schlegel et al. 2021) adapt LIME to forecasting, testing several segmentation algorithms and simple perturbation strategies. For the sake of simplicity, we choose to use their work to perform our LIME analysis, focusing on the baseline segmentation algorithm, a uniform segmentation of the time-series, with perturbation strategies chosen for each model.

2.4 Rating of AI Models

Prior work has developed rating methods to assess AI models from a third-party perspective, often without access to training data. Early work (Srivastava and Rossi 2020) measured bias, such as gender bias in machine translation (Srivastava and Rossi 2018), and used visualizations to communicate results (Bernagozzi et al. 2021b,a), but lacked a causal foundation. More recent approaches introduced causal rating methods to isolate the effect of protected attributes on model outputs. These were applied first to sentiment analysis (Lakkaraju, Srivastava, and Valtorta 2024), later to composite NLP tasks (Lakkaraju et al. 2023), and then to time-series forecasting (Lakkaraju et al. 2024, 2025). In this work, we build on that foundation, using ratings not only as performance metrics, but also as Rating-Driven Explanations (RDE), originally introduced in (Lakkaraju, Valluru, and Srivastava 2025). This allows us to describe how models behave when sensitive attributes vary, complementing traditional XAI methods that focus on individual predictions or global feature importance.

3 Problem Setting

3.1 Mathematical Formulation

Let $\mathcal{D} = D_1, D_2, D_3, D_4$ denote our collection of four datasets from distinct domains, each described in detail in Section 4.1. For each dataset D_i , we observe a univariate time-series $\mathbf{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_{T_i}^{(i)}\}$ where T_i represents the total length of series i . While three datasets consist of continuous-time observations, the financial dataset comprises business daily observations (excluding weekends and holidays), which we treat as continuous for the purpose of forecasting.

3.2 Forecasting Task

Given a time-series $\mathbf{x}^{(i)}$ of length T_i , we partition the data into training and test sets using an 80-20 split:

$$\mathbf{x}_{\text{train}}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_{\lfloor 0.8T_i \rfloor}^{(i)}\} \quad (1)$$

$$\mathbf{x}_{\text{test}}^{(i)} = \{x_{\lfloor 0.8T_i \rfloor + 1}^{(i)}, \dots, x_{T_i}^{(i)}\} \quad (2)$$

The forecasting methodology varies by model architecture. Traditional models (ARIMA, Gradient Boosting) employ iterative, single-step forecasting to cover the entire test set. For models that benefit from fixed context and horizon windows (Chronos, LLaMA, LLaMA-FT), we define domain-specific windowed forecasting tasks:

1. Finance: Given context length $C = 20$ and horizon $H = 5$, predict 5 business days using the preceding 4 weeks of adjusted closing prices.

2. Power: Given context length $C = 1440$ and horizon $H = 360$, predict 6 hours of minutely power levels using the previous day’s data.

3. Pedestrian: Given context length $C = 72$ and horizon $H = 18$, predict 18 hours of pedestrian counts using the previous 3 days of hourly data.

4. Car: Given context length $C = 8$ and horizon $H = 2$, predict 2 months of car part sales using the previous 8 months of data.

For windowed models, let $\mathbf{x}_{t-C+1:t}^{(i)} = \{x_{t-C+1}^{(i)}, \dots, x_t^{(i)}\}$ denote the context window and $\hat{\mathbf{x}}_{t+1:t+H}^{(i)} = \{\hat{x}_{t+1}^{(i)}, \dots, \hat{x}_{t+H}^{(i)}\}$ the predicted horizon. The forecasting process slides this window across the test set to generate predictions for the entire test period.

4 Methods

We describe the implementation and evaluation procedure used to generate forecasts, interpret model behavior, and compute rating-based explanations. All forecasting experiments follow the unified data split and rolling forecast protocol defined in Section 3.

4.1 Data Description

We use the following datasets for training and evaluating the TSFM, where T_i denotes the length of the series as defined in Section 3.

1. Finance (D_1): Six parallel financial time-series ($T_i = 250$ for all series) representing daily closing prices from March 28, 2023, to March 22, 2024 (ensuring no overlap with Chronos’s training data). Series are evenly split between technology, finance, and pharmaceutical sectors.

2. Power (D_2): A single univariate series ($T_i = 20,915$) of minute-level energy consumption measurements from Tantiv4 (Muppasani et al. 2023).

3. Pedestrian (D_3): 66 hourly pedestrian count series with varying lengths ($576 \leq T_i \leq 96,424$) from Melbourne sensors (City of Melbourne 2024).

4. Car Parts (D_4): 2,674 monthly sales series ($T_i = 51$ for all series) with extreme sparsity (75.9% zero values) (Hyndman et al. 2008).

We preprocess each dataset into a consistent long-format schema with columns `series_id`, `timestamp`, and `value`. We make no assumptions regarding stationarity, ergodicity, or other statistical properties of the time-series. All datasets are characterized by inherent forecasting difficulty, representing challenging real-world scenarios where traditional statistical assumptions may not hold. These datasets collectively capture a range of forecasting scenarios.

4.2 Forecasting Procedure

To generate predictions beyond each model’s native horizon, we adopt an **iterative forecasting** strategy. Models first predict for a window of length equal to their native horizon; these predictions are appended to the historical context and used to forecast the next window. This process repeats until the test period is fully covered. The full prediction history is retained for each series, and only the required number of steps is generated at each iteration to avoid compounding unnecessary error. Two inference setups are used:

Autoregressive: Iterative rollout until the forecast window is covered.

Direct: A single context window is used to predict the entire horizon in one shot.

For Llama and its fine-tuned version, we use the direct setup on the Power and Pedestrian datasets due to their large size, which makes full autoregressive rollouts impractical within our time budget. All other datasets use the autoregressive setup. Domain-specific context (C) and horizon (H) lengths are detailed in Section 3. During training, sliding windows are extracted from the training series. During inference, these windows slide across the test period to cover the full horizon.

4.3 Models

We evaluate four models spanning statistical baselines, feature-based ML, and foundation models (Table 2).

1. ARIMA: Implemented with `statsmodels` per series. Data are scaled with a `RobustScaler` and inverse-transformed after forecasting. Model orders (p, d, q) are selected via grid search constrained by domain-specific maxima; d is chosen using the Augmented Dickey–Fuller and KPSS tests. Seasonal orders (P, D, Q, s) are included when $s > 1$ and data length permits, with s inferred from autocorrelation peaks or domain defaults (e.g., $s = 5$ for finance, $s = 1440$ for power). Grid search considers $p, q \leq 5$, $d \in \{0, 1, 2\}$. Forecasting is iterative single-step with a rolling window ($w = 7$ for finance, $w = 1440$ for power). Confidence intervals (95%) are used for uncertainty quantification.

2. Gradient Boosting: Implemented with LightGBM (`lgb.LGBMRegressor`) using engineered features: time-based (hour, day, month), Fourier terms for seasonality, domain-specific statistics (e.g., log returns, volatility, moving averages), and tailored lags. Models are trained independently on the training set only (500 estimators, learning rate 0.05, L1 loss). Forecasting is iterative, with features recalculated after each prediction.

3. Chronos: A pretrained transformer for time-series, implemented via AutoGluon TimeSeries in “bolt_small” configu-

ration. Data are resampled to known frequencies per domain (hourly, monthly, minutely, business days), using forward-fill interpolation except for car sales, which uses zero-fill to preserve sparsity. All numeric values are cast to float32.

4. Llama/Llama-FT: Meta-Llama-3.1-8B-Instruct fine-tuned with LoRA adapters on MinMax-scaled windows, where each prompt contains C past values and H future targets. LoRA configuration: $r = 16$, $\alpha = 32$, dropout = 0.05; quantization: NF4, bfloat16; optimizer: paged AdamW with cosine LR decay, gradient accumulation, and fp16. We refer to the fine-tuned variant as Llama-FT and the original base model as Llama.

4.4 Evaluation Metrics

Model accuracy is evaluated with:

$$\text{sMAPE} = \frac{100}{n} \sum_{t=1}^n \frac{|x_t - \hat{x}_t|}{(|x_t| + |\hat{x}_t|)/2}$$

$$\text{MASE} = \frac{\frac{1}{n} \sum_{t=1}^n |x_t - \hat{x}_t|}{\frac{1}{m-s} \sum_{t=s+1}^m |x_t - x_{t-s}|}$$

where n is the test length, m the training length, and s the seasonal period ($s = 1440$ power, $s = 24$ pedestrian, $s = 5$ finance, $s = 1$ car). Metrics are computed per series and aggregated by domain.

4.5 Explainability Analysis

We use SHAP to compute feature attributions. For GBoost, TreeSHAP is applied directly. For Chronos, surrogate LightGBM models approximate outputs for SHAP analysis (Raykar et al. 2023), with fidelity checked via RMSE. TS-MULE (a LIME variant for time-series) is also used, perturbing uniform segments with mean or zero replacement to explain the final forecast value.

4.6 Rating-Driven Explanations (RDE)

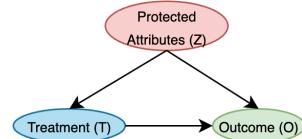


Figure 1: Proposed Generalized Causal Graph.

The RDE framework (Lakkaraju, Valluru, and Srivastava 2025) integrates causal reasoning with metrics to explain model behavior under different conditions. The generalized causal graph used for our experiments is shown in Figure 1. A causal model specifies relationships between treatment T , outcome O , and protected attribute Z . If Z confounds T and O , deconfounding is required. We focus on:

- **Weighted Rejection Score (WRS):** Measures statistical differences in O across Z groups via multiple t -tests, weighted by significance level. Higher WRS = more Z sensitivity.

- **Average Treatment Effect (ATE):** Estimates the causal effect of T on O using G-computation (Robins 1986) to account for confounding.

More details on the metrics are provided in Section A in the appendix. The workflow (Figure A.1) takes a user query or a hypothesis (as shown in Tables 4 and 5), identifies T , O , and Z , selects a metric, computes metrics, and compares them to random and biased baselines to contextualize findings.

5 Experiments and Results

In this section, we detail the experimental setup used to address the research questions stated in Section 1.

5.1 RQ1: When forecasting models succeed or fail?

Experimental Setup: To answer this question, we collected complementary performance metrics sMAPE and MASE through forecasting experiments as detailed in Sections 3 and 4.

Results: Table 3 summarizes forecasting performance across four domains using MASE and sMAPE, with lower values indicating better accuracy. Gradient Boosting (GBoost) generally achieved the strongest overall performance, obtaining the lowest MASE in finance (4.32 ± 3.00), power (0.84 ± 0.00), pedestrian (1.00 ± 0.39), and car parts (0.77 ± 1.32). Chronos performed competitively in certain cases, delivering the best sMAPE for finance ($5.30 \pm 5.15\%$) and strong results for power (8.84%), though its pedestrian accuracy lagged behind GBoost. ARIMA yielded reasonable performance in some domains but was consistently outperformed by GBoost, particularly in power and pedestrian datasets where the latter reduced MASE by more than half.

The LLaMA model, without fine-tuning, struggled across domains, producing large errors, especially in finance (MASE = 88.66) and pedestrian datasets, while fine-tuning (LLaMA-FT) reduced errors substantially but did not surpass GBoost or Chronos in any metric. In the car parts domain, only GBoost and ARIMA results are available for MASE due to Chronos limitations, while sMAPE is missing for ARIMA and GBoost. Across all domains, the standard deviation of results highlights the variability in model stability, with GBoost consistently showing both strong mean performance and relatively low variance, reinforcing its robustness across heterogeneous time-series forecasting tasks.

Interpretation: Forecasting models succeed or fail based on how well their assumptions and architectures align with the data’s statistical and structural properties. Gradient Boosting excels across most domains, leveraging engineered time-based and domain-specific features to overcome challenges like inconsistent seasonality, sparsity, or long sequences. In contrast, ARIMA performs well on short, stationary series with strong autocorrelations but struggles with high-frequency or nonstationary data (e.g., minute-level power consumption) due to its linearity. Chronos generalizes well, achieving strong sMAPE in finance and competitive results in power. However, like other models, its advantage diminishes with irregular seasonality or extreme sparsity (e.g.,

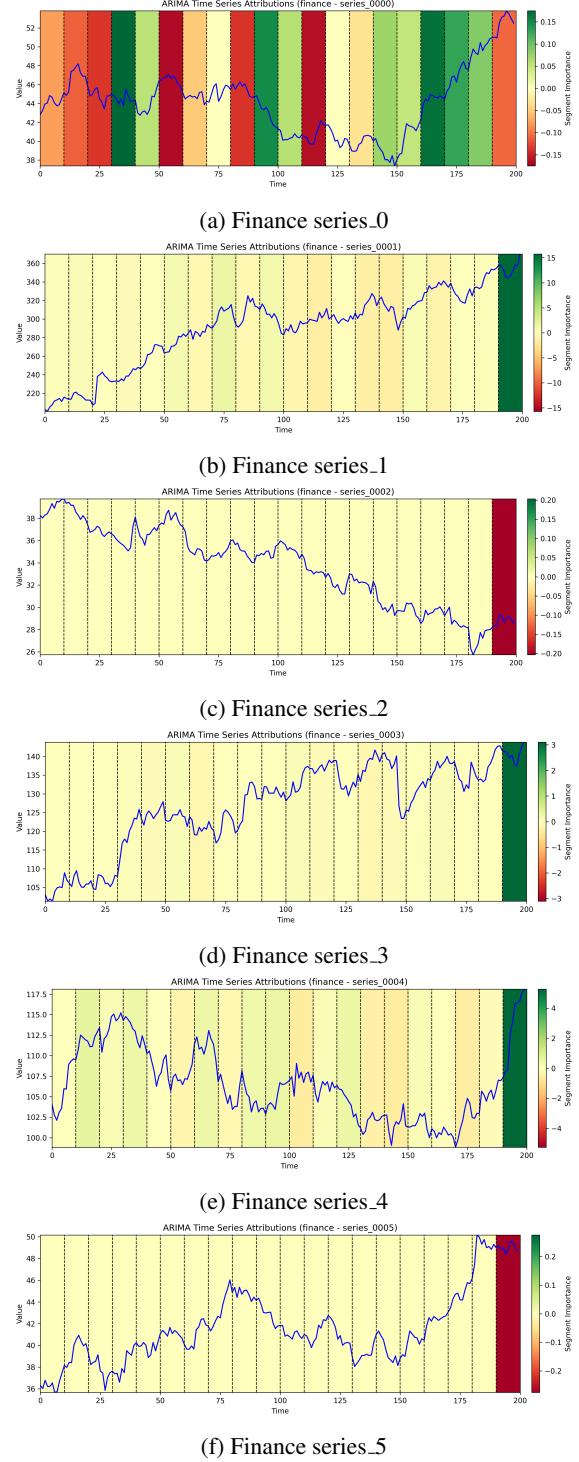


Figure 2: LIME plots for ARIMA model in finance using 20 uniform segments, local-mean replacement, and 200 samples. Figures show the distribution of feature importance for each time-series. Panels (c) and (f) show very little feature importance (< 0.000001) assigned to values to before the last segment, while panels (a), (b), (d), and (e) show varying feature importance in all segments.

Domain	Metric	ARIMA	GBoost	Chronos*	LLaMa	LLaMa-FT
Finance	MASE	7.69 ± 6.56	4.32 ± 3.00	5.48 ± 5.44	88.06 ± 23.88	8.92 ± 6.13
	sMAPE	5.90 ± 6.37	10.27 ± 7.37	5.30 ± 5.15	169.03 ± 47.91	8.64 ± 5.99
Power	MASE	2.89 ± 0.00	0.84 ± 0.00	2.33 ± 0.00	40.06 ± 0.00	14.07 ± 0.00
	sMAPE	22.08 ± 0.00	6.86 ± 0.00	8.84 ± 0.00	17.75 ± 0.00	6.04 ± 0.00
Pedestrian	MASE	1.86 ± 7.77	1.00 ± 0.39	$1.47 \pm 0.67^*$	3.21 ± 1.57	1.77 ± 0.77
	sMAPE	214.41 ± 27.2	50.64 ± 11.78	$90.98 \pm 44.77^*$	107.94 ± 61.36	72.85 ± 13.46
Car Parts	MASE	1.38 ± 3.40	0.77 ± 1.32	∞	∞	∞
	sMAPE	∞	49.97 ± 52.80	$175.03 \pm 37.56^*$	142.76 ± 89.46	138.53 ± 89.04

Table 3: Forecasting performance comparison. MASE and sMAPE are reported as (mean \pm std. dev.). sMAPE is reported as percentage (%). GBoost \rightarrow Gradient Boosting. Best results are in **bold** (lower value is better). **Across domains, GBoost exhibits best MASE and sMAPE with the exception of sMAPE finance.** *During the initial training and evaluation of Chronos (Ansari et al. 2024), the model was exposed to the pedestrian and car parts dataset in an experimental setup different than our own.

car parts, pedestrian datasets), where feature-driven methods like Gradient Boosting outperform it. General-purpose FM, Llama, is highly sensitive to sequence length, sparsity, and distributional shifts; without fine-tuning, it performs poorly. While Llama’s low sMAPE suggests good proportional accuracy, its high MASE reveals larger absolute errors than the naive baseline, which performed well due to minimal inter-step variation (Table 3).

Conclusion: Ultimately, model success hinges on the alignment between inductive bias and dataset characteristics (frequency, length, regularity, sparsity, seasonality). Domain knowledge integration remains the most critical factor in time-series forecasting.

5.2 RQ2: What do common XAI methods tell us about why forecasting models succeed or fail?

Experimental Setup: To answer this question, we applied variations of SHAP and LIME as detailed in sections 3 and 4.

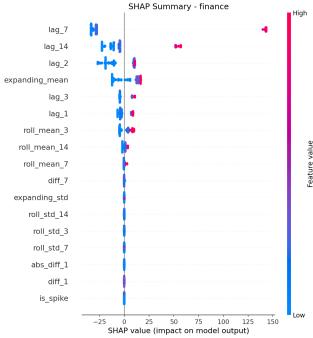


Figure 3: SHAP values for the Chronos Surrogate model (finance domain). The full set of domain-specific results is provided in Figure B.2 (Appendix).

TsSHAP - Chronos Surrogate Model: In the **Finance** domain, lag features comprise 4 of the top 5 features, with expanding mean being the only non-lag feature. This, combined with the Surrogate’s performance metrics found in

the appendix Table B.1 shows that Chronos and its surrogate captures trends despite no domain-specific features. The plot presents a clear directional pattern: high historical values boost predictions, low values suppress them. Surrogate maintains GBoost’s lag sensitivity but replaces engineered features with statistical aggregates.

Interpretation: When placed in conversation with the other domains found in Table B.2 (in appendix), Chronos’s surrogate reveals a fundamental tension, while capable of detecting simple trends (finance lags), they default to statistical baselines (means/standard deviations) when faced with complex patterns. This suggests pretrained models may *recognize* temporal structures but struggle to *leverage* them without domain-specific feature engineering. The surrogate’s performance floor becomes evident in challenging domains (car/power), where it reverts to predicting averages rather than meaningful values.

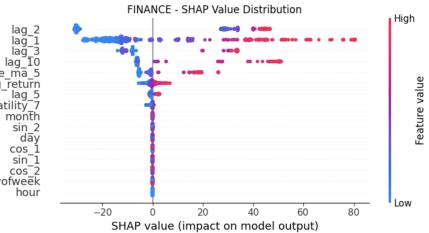


Figure 4: SHAP values for the Gradient Boosting model (finance domain). The full set of domain-specific results is provided in Figure B.3 (Appendix).

TreeSHAP - Gradient Boosting: In **Finance**, lagged values constitute 4 of the top 5 features, alongside a 5-day moving average. This confirms the trend-driven nature of financial markets, where price movements exhibit momentum effects. Calendar features (day/month) showed negligible importance, suggesting market reactions depend more on events than temporal context. The dominance of engineered features over raw lags highlights how domain-specific transformations can capture meaningful patterns.

Interpretation: In the other 3 domains (Appendix) we see

a similar pattern, while lagged values provide a universal foundation, each domain’s unique physics (market trends, human schedules, periodicity, or sparsity patterns) dictate which features ultimately drive predictive power.

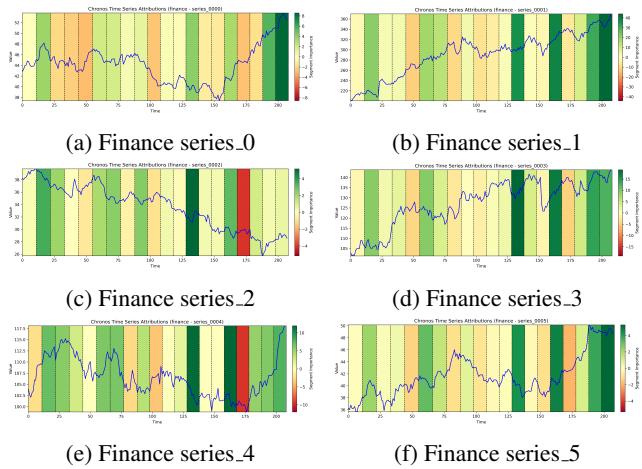


Figure 5: LIME plots for Chronos model in finance using 20 uniform segments, zero replacement, and 200 samples. Figures show the distribution of feature importance for each time-series, revealing a relatively balanced distribution, with slight bias towards the later segments.

Segment Based LIME - Chronos: In Finance, figure 5 clearly shows that earlier points in the time-series have significant effects on the final value of the forecast, indicating that Chronos is picking up on patterns in the data and using them to determine the direction of the overall forecast. Since zero-replacement is being used, it is expected that all final segments contribute positively towards the final forecast value. A clear pattern emerged with high feature importance being placed on the segments corresponding the 130-140th and 150-160th points in the time-series. The segments which bring down the value of the final forecast do not follow any obvious patterns, indicating that this explanation method may be a better diagnostic tool, rather than a tool for informed decision making.

Segment Based LIME - ARIMA: The LIME explanations for ARIMA in finance clearly demonstrate its strong dependence on the most recent time segments, which aligns with the model’s autoregressive structure. This behavior is inherent to ARIMA’s design - the autoregressive component explicitly prioritizes recent observations when generating forecasts, creating the characteristic pattern of exponentially decaying influence as we look further back in time. Figures 2c and 2f show that for the last time point in the forecast, none of the segments other than the last had much of an effect on the output. Visually, figure 2a looks like an outlier, but only shows a lower attribution to the final segment. The values of all other segments are consistent with the distributions seen in figures 2b, 2d, and 2e.

Dataset	Model	ATE (lower is better)	Rating
Cars	GBoost	0.22	1
	ARIMA	0.26	2
	Llama-FT	0.46	3
	Llama	0.47	4
Pedestrian	GBoost	320.50	1
	Llama-FT	500.30	2
	Llama	975.48	3
	ARIMA	2803.08	4
Finance	Chronos	13.64	1
	ARIMA	18.19	2
	Llama-FT	20.20	3
	GBoost	28.76	4
	Llama	106.02	5

Table 4: Hypothesis 1: After accounting for the protected attribute, the model makes different levels of error depending on which series (car part, company, or sensor) is being predicted. This shows whether the model is consistent across all series or performs better for some than others.

Dataset	Model	WRS (lower is better)	Rating
Cars	ARIMA	0.56	1
	Llama-FT	0.60	2
	Llama	0.63	3
	GBoost	0.66	4
Pedestrian	Llama-FT	0.39	1
	Llama	0.49	2
	ARIMA	0.83	3
	GBoost	0.87	4
Finance	Llama	0.27	1
	Chronos	0.66	2
	ARIMA	0.75	3
	Llama-FT	0.85	4
	GBoost	0.85	4

Table 5: Hypothesis 2: Residual distributions differ across protected groups, meaning the model makes systematically different errors depending on the month of the year (cars, finance) or the day of the week (pedestrian). This matters because such sensitivity can cause the model to be reliable in some periods but noticeably less accurate in others, leading to uneven performance for the user.

5.3 RQ3: How can rating augment our understanding of when and why forecasting models succeed or fail?

Experimental Setup: We analyze the same models as in Table 3 (ARIMA, GBoost, Chronos, LLaMA, LLaMA-FT). Accuracy is reported using *MASE* and *sMAPE*. To examine how errors vary, we compute two RDE metrics, previously defined in Section 4.6, on the Cars, Pedestrian, and Finance datasets. The Power dataset was excluded from the RDE analysis as it contains only a single series with minutely observations, making the definition of a meaningful protected attribute infeasible:

ATE (Table 4): Estimated effect of the *series identifier* (T) on the residual (O), after adjusting for the protected attribute (Z): month for Cars/Finance, day-of-week for Pedestrian. Higher values indicate greater series-specific error (e.g., part type in Cars, company in Finance, sensor in Pedestrian).

WRS (Table 5): Measures differences in residual distributions across protected groups (Z): month for Cars/Finance, day-of-week for Pedestrian. Lower values indicate less sen-

sitivity of errors to these groups.

Interpretation: GBoost’s strong accuracy (Table 3) coincides with *low ATE* in Cars and Pedestrian (Table 4), errors are relatively uniform across series after adjustment, but *high WRS* across all three datasets (Table 5). **Thus, GBoost can be accurate on average yet seasonally/day-wise sensitive, producing uneven errors across months or weekdays.** In Finance, Chronos achieves the lowest ATE (errors consistent across series) but mid-range WRS, while Llama attains the lowest WRS (errors consistent across months/days) but weak ATE and lower accuracy, different models control different sources of variability. The SHAP summaries (Figures B.2, B.3) show heavy reliance on lag/expanding statistics; LIME plots (Figures 5, 2) often emphasize later segments of the context window. Ratings translate these attribution patterns into measurable consequences: when a model relies on features that co-vary with month/day, **WRS increases**; when it overfits to series-specific patterns, **ATE increases**.

Conclusion: Ratings provide the missing link between forecasting accuracy and feature attributions. ATE captures *series-dependent error* after adjusting for potential confounders, while WRS captures *error differences across months or days*. Together with Table 3, they explain why a model “performs well” or “fails” in a domain: for example, GBoost’s strong accuracy comes from low ATE but it fails stability checks across time groups (high WRS). These diagnostics support domain-aware model selection, focusing on the type of instability: series vs. month/day, that the ratings expose.

6 Discussion and Conclusion

This paper investigated three core research questions to evaluate the performance, interpretability, and robustness of forecasting models across diverse time-series domains. Our experiments and analyses yielded the following key insights:

6.1 Summary of Findings

RQ1: When do forecasting models succeed or fail? Model performance is determined by the alignment between their inductive biases and the statistical properties of the data. GBoost demonstrated superior performance, achieving the lowest MASE across all domains by leveraging domain-specific feature engineering to handle challenges such as sparsity, non-stationarity, and irregular seasonality. In contrast, ARIMA performed competitively only on short, near-stationary series with strong autocorrelations, while Chronos excelled in finance but struggled in domains with extreme sparsity (e.g., car parts). The base Llama model showed degradation in forecasting accuracy, revealing that without domain-specific fine-tuning it cannot generalize to temporal patterns in the target data and is highly vulnerable to shifts between training and test distributions.

RQ2: What do XAI methods reveal about model behavior? SHAP and LIME analyses provided mechanistic explanations for model successes and failures: GBoost relied heavily on engineered features (even if these features are as simple as domain-specific lags), confirming its adaptability

to domain-specific temporal structures. Chronos’s Surrogate revealed a tendency to default to statistical aggregates (e.g., expanding means) in complex domains, indicating its inability to leverage intricate patterns without explicit feature engineering. ARIMA’s assumptions, per LIME, led to over-reliance on recent lags, while LLaMA’s poor fine-tuning performance highlighted its dependence on pretraining data alignment.

RQ3: How can rating metrics augment our understanding? The Average Treatment Effect (ATE) and Weighted Rejection Score (WRS) metrics quantified two critical dimensions of model reliability: Series-specific consistency (ATE): GBoost’s low ATE in pedestrian and car datasets indicated uniform errors across series, while Chronos’s low ATE in finance validated its generalization capability. Temporal stability (WRS): GBoost’s high WRS revealed seasonal and day-of-week error variance, suggesting that its average accuracy may mask periodic unreliability. While our results highlight the promise of Rating-Driven Explanations (RDE), they are not directly compared against other fairness-oriented XAI approaches. Similarly, we did not evaluate RDE on synthetic benchmarks with ground-truth fairness properties. Both comparisons would provide a stronger test of our third research question and are left for future work.

6.2 Implications and Future Work

Our findings show that effective forecasting depends on how well the modeling approach matches the structure and constraints of the domain. GBoost consistently outperforms other models when domain-specific features can be engineered (echoing the findings of (Elsayed et al. 2021)), demonstrating the strength of informed feature design over learned representations. Chronos and Llama-FT, while capable in domains with trend-driven data (e.g., finance), performed inconsistently in sparse or irregular settings (e.g., car sales). Significant opportunities exist to advance explainability in time-series forecasting. LIME adaptations could be more thoroughly explored, while this paper evaluates a baseline implementation (TS-MULE with uniform segmentation), prior work in classification has demonstrated more advanced techniques (e.g., dynamic segmentation or saliency-based sampling (Sivill and Flach 2022)) that may improve explanation stability for temporal data. Additionally, expanding the evaluation of forecast outputs beyond point predictions, such as analyzing the first predicted value, midpoint of the forecast horizon, or derived features (e.g., maximum/minimum values), could reveal new insights into model behavior across different temporal contexts. Third, incorporating physics-informed constraints into temporal foundation models (TFMs) may improve their ability to capture domain-specific dynamics while maintaining interpretability. Additionally, the proposed rating metrics (ATE/WRS) should be rigorously evaluated as XAI methods themselves, following frameworks like (Mersha et al. 2024) to quantify their effectiveness in surfacing model biases. Finally, combining these directions could yield unified evaluation protocols that assess models holistically across accuracy, robustness, and explainability dimensions.

References

- Abbasimehr, H.; and Paki, R. 2022. Improving time series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing*, 13(1): 673–691.
- Agapitos, A.; Brabazon, A.; and O’Neill, M. 2017. Regularised gradient boosting for financial time-series modelling. *Computational Management Science*, 14(3): 367–391.
- Alfeo, A. L.; Zippo, A. G.; Catrambone, V.; Cimino, M. G.; Toschi, N.; and Valenza, G. 2023. From local counterfactuals to global feature importance: efficient, robust, and model-agnostic explanations for brain connectivity networks. *Computer Methods and Programs in Biomedicine*, 236: 107550.
- Alikhademi, K.; Richardson, B.; Drobina, E.; and Gilbert, J. E. 2021. Can explainable AI explain unfairness? A framework for evaluating explainable AI. *arXiv preprint arXiv:2106.07483*.
- Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.
- Arrieta, A. B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58: 82–115.
- Arsenault, P.-D.; Wang, S.; and Patenaude, J.-M. 2025. A survey of explainable artificial intelligence (XAI) in financial time series forecasting. *ACM Computing Surveys*, 57(10): 1–37.
- Bernagozzi, M.; Srivastava, B.; Rossi, F.; and Usmani, S. 2021a. Gender Bias in Online Language Translators: Visualization, Human Perception, and Bias/Accuracy Tradeoffs. *IEEE Internet Computing*, 25(5): 53–63.
- Bernagozzi, M.; Srivastava, B.; Rossi, F.; and Usmani, S. 2021b. Vega: a virtual environment for exploring gender bias vs. accuracy trade-offs in ai translation services. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15994–15996.
- Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.
- Çelik, T. B.; İcan, Ö.; and Bulut, E. 2023. Extending machine learning prediction capabilities by explainable AI in financial time series prediction. *Applied Soft Computing*, 132: 109876.
- City of Melbourne. 2024. Pedestrian Counting System. <https://www.melbourne.vic.gov.au/pedestrian-counting-system>. Accessed: 2024-08-10.
- Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2023. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*.
- Ekambaram, V.; Jati, A.; Dayama, P.; Mukherjee, S.; Nguyen, N. H.; Gifford, W. M.; Reddy, C.; and Kalagnanam, J. 2024. Tiny Time Mixers (TTMs): Fast Pre-trained Models for Enhanced Zero/Few-Shot Forecasting of Multivariate Time Series. *CoRR*.
- Elsayed, S.; Thyssens, D.; Rashed, A.; Schmidt-Thieme, L.; and Jomaa, H. S. 2021. Do We Really Need Deep Learning Models for Time Series Forecasting? *CoRR*, abs/2101.02118.
- Garza, A.; and Mergenthaler-Canseco, M. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589*.
- Godahewa, R.; Bergmeir, C.; Webb, G. I.; Hyndman, R. J.; and Montero-Manso, P. 2021. Monash Time Series Forecasting Archive. *arXiv:2105.06643*.
- Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.
- Greenwell, B. M. 2017. pdp: An R Package for Constructing Partial Dependence Plots. *The R Journal*, 9(1): 421–436.
- Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.
- Hoffman, R. R.; Mueller, S. T.; Klein, G.; Jalaeian, M.; and Tate, C. 2023. Explainable ai: roles and stakeholders, desirments and challenges. *Frontiers in Computer Science*, 5: 1117848.
- Hyndman, R.; Koehler, A.; Ord, K.; and Snyder, R. 2008. *Forecasting with exponential smoothing: the state space approach*. Springer.
- Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.
- Lakkaraju, K.; Gupta, A.; Srivastava, B.; Valtorta, M.; and Wu, D. 2023. The Effect of Human v/s Synthetic Test Data and Round-Tripping on Assessment of Sentiment Analysis Systems for Bias. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 380–389. Los Alamitos, CA, USA: IEEE Computer Society.
- Lakkaraju, K.; Kaur, R.; Zehtabi, P.; Patra, S.; Valluru, S. L.; Zeng, Z.; Srivastava, B.; and Valtorta, M. 2025. On Creating a Causally Grounded Usable Rating Method for Assessing the Robustness of Foundation Models Supporting Time Series. *arXiv preprint arXiv:2502.12226*.
- Lakkaraju, K.; Kaur, R.; Zeng, Z.; Zehtabi, P.; Patra, S.; Srivastava, B.; and Valtorta, M. 2024. Rating Multi-Modal Time-Series Forecasting Models (MM-TSFM) for Robustness Through a Causal Lens. *arXiv preprint arXiv:2406.12908*.
- Lakkaraju, K.; Srivastava, B.; and Valtorta, M. 2024. Rating Sentiment Analysis Systems for Bias Through a Causal Lens. *IEEE Transactions on Technology and Society*, 1–1.

- Lakkaraju, K.; Valluru, S. L.; and Srivastava, B. 2025. Holistic Explainable AI (H-XAI): Extending Transparency Beyond Developers in AI-Driven Decision Making. *arXiv*:2508.05792.
- Li, Y.; Zhu, Z.; Kong, D.; Han, H.; and Zhao, Y. 2019. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems*, 181: 104785.
- Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Mersha, M.; Lam, K.; Wood, J.; AlShami, A. K.; and Kalita, J. 2024. Explainable artificial intelligence: A survey of needs, techniques, applications, and future direction. *Neurocomputing*, 599: 128111.
- Minarso, V. P.; Adjji, T. B.; and Setiawan, N. A. 2022. Hybrid SVD-ARIMA Method for Sales Forecasting with Sparse Data on E-Commerce Products. In *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 387–392.
- Mokhtari, K. E.; Higdon, B. P.; and Başar, A. 2019. Interpreting financial time series with SHAP values. In *Proceedings of the 29th annual international conference on computer science and software engineering*, 166–172.
- Muppasani, B.; Anand, C. J.; Appajigowda, C.; Srivastava, B.; and Johri, L. 2023. A dataset and baseline approach for identifying usage states from non-intrusive power sensing with MiDAS IoT-based sensors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15545–15550.
- Pan, Z.; Jiang, Y.; Garg, S.; Schneider, A.; Nevyvaka, Y.; and Song, D. 2024. IP-LLM: Semantic space informed prompt learning with LLM for time series forecasting. In *Forty-first International Conference on Machine Learning*.
- Rasul, K.; Ashok, A.; Williams, A. R.; Khorasani, A.; Adamopoulos, G.; Bhagwatkar, R.; Biloš, M.; Ghonia, H.; Hassen, N. V.; Schneider, A.; et al. 2023. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*.
- Raykar, V. C.; Jati, A.; Mukherjee, S.; Aggarwal, N.; Sarpatwar, K.; Ganapavarapu, G.; and Vaculin, R. 2023. TsSHAP: Robust model agnostic feature-based explainability for time series forecasting. *arXiv preprint arXiv:2303.12316*.
- Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv*:1602.04938.
- Robins, J. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12): 1393–1512.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5): 688.
- Schlegel, U.; Arnout, H.; El-Assady, M.; Oelke, D.; and Keim, D. A. 2019. Towards a rigorous evaluation of XAI methods on time series. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 4197–4201. IEEE.
- Schlegel, U.; Lam, D. V.; Keim, D. A.; and Seebacher, D. 2021. TS-MULE: Local Interpretable Model-Agnostic Explanations for Time Series Forecast Models. *arXiv*:2109.08438.
- Shih, S.-Y.; Sun, F.-K.; and Lee, H.-y. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108: 1421–1441.
- Shumway, R. H.; and Stoffer, D. S. 2017. ARIMA models. In *Time series analysis and its applications: with R examples*, 75–163. Springer.
- Sivill, T.; and Flach, P. 2022. LIMESegment: Meaningful, Realistic Time Series Explanations. In Camps-Valls, G.; Ruiz, F. J. R.; and Valera, I., eds., *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, 3418–3433. PMLR.
- Splawa-Neyman, J.; Dabrowska, D. M.; and Speed, T. P. 1990. On the application of probability theory to agricultural experiments. Essay on principles. Section 9. *Statistical Science*, 465–472.
- Srivastava, B.; and Rossi, F. 2018. Towards Composable Bias Rating of AI Systems. In *2018 AI Ethics and Society Conference (AIES 2018), New Orleans, Louisiana, USA, Feb 2-3*.
- Srivastava, B.; and Rossi, F. 2020. Rating AI Systems for Bias to Promote Trustable Applications. In *IBM Journal of Research and Development*.
- Student. 1908. The probable error of a mean. *Biometrika*, 1–25.
- Talukder, S.; Yue, Y.; and Gkioxari, G. 2024. TOTEM: TOkenized Time Series EMbeddings for General Time Series Analysis. *arXiv preprint arXiv:2402.16412*.
- Tasin, I.; Nabil, T. U.; Islam, S.; and Khan, R. 2023. Diabetes prediction using machine learning and explainable AI techniques. *Healthcare technology letters*, 10(1-2): 1–10.
- Tire, K.; Taga, E. O.; Ildiz, M. E.; and Oymak, S. 2024. Retrieval-augmented time series forecasting. *arXiv preprint arXiv:2411.08249*.
- Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*.
- Xiao, M.; Jiang, Z.; Qian, L.; Chen, Z.; He, Y.; Xu, Y.; Jiang, Y.; Li, D.; Weng, R.-L.; Peng, M.; et al. 2025. Retrieval-augmented large language models for financial time series forecasting. *arXiv preprint arXiv:2502.05878*.
- Yu, X.; Chen, Z.; Ling, Y.; Dong, S.; Liu, Z.; and Lu, Y. 2023. Temporal Data Meets LLM—Explainable Financial Time Series Forecasting. *arXiv preprint arXiv:2306.11025*.
- Zhang, Y.; Petrosian, O.; Liu, J.; Ma, R.; and Krinkin, K. 2022. FI-SHAP: explanation of time series forecasting and improvement of feature engineering based on boosting algorithm. In *Proceedings of SAI intelligent systems conference*, 745–758. Springer.
- Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.

Appendix

A RDE

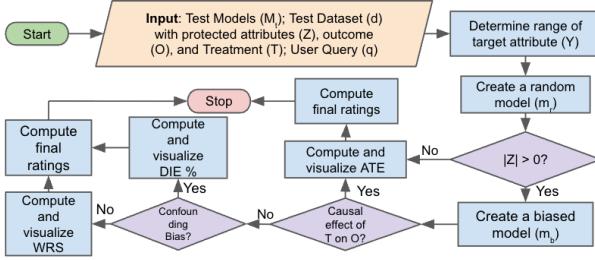


Figure A.1: RDE Workflow.

Although (Lakkaraju, Valluru, and Srivastava 2025) also considered another rating metric, DIE %, we restrict our analysis to two metrics, ATE and WRS, to validate the hypotheses in Tables 4 and 5:

Weighted Rejection Score (WRS): WRS quantifies statistical bias across a protected attribute $z_i \in Z$ by comparing outcome distributions ($O | z_i$) across all $\binom{m}{2}$ group pairs using Student's t-test (Student 1908). For each pair, the t-value t_{z_i} is compared to the critical value t_{crit} for a given confidence interval (CI). A rejection indicates a significant difference in outcomes. WRS is computed as a weighted sum of rejections across CIs, where x_i is the number of rejections and w_i is the assigned weight (e.g., 1, 0.8, 0.6 for 95%, 75%, and 60% CIs). Formally, $WRS = \sum_i w_i \cdot x_i$.

Average Treatment Effect (ATE): Average Treatment Effect (ATE) (Rubin 1974; Splawa-Neyman, Dabrowska, and Speed 1990) measures the average difference in outcomes between the treated ($T = i$) and untreated ($T = 0$) units, quantifying the impact of the treatment on the AI model's outcome. Unlike local counterfactuals, ATE quantifies the average effect of a treatment on model predictions across the dataset, making it well-suited for answering stakeholder questions about overall effect of an attribute on the outcome. Formally, $ATE = [|E[O|do(T = i)] - E[O|do(T = 0)]|]$

B SHAP Analysis

In **Power**, statistical features outperform lags: all rolling means rank above any lag feature. Expanding standard deviation has a strong negative impact - high volatility triggers downward predictions. Power is a volatile domain where “default to mean” becomes dominant strategy. From Figure B.2, we see that in the **Pedestrian** domain, lags make up 2 of the top 5 features. Once again, expanding std. emerges as the most important. The other important statistical features were expanding mean and the difference between the last time pt and lag 7. Rolling means prove irrelevant. Given the size of the pedestrian dataset and the lack of feature engineering for the surrogate model it is obvious that the short lags fail to capture the daily and weekly seasonality. In the **Car** domain, expanding mean dominates completely - model collapses to baseline in sparse domain. High feature values correlate linearly with SHAP values, indicating limited pattern capture.

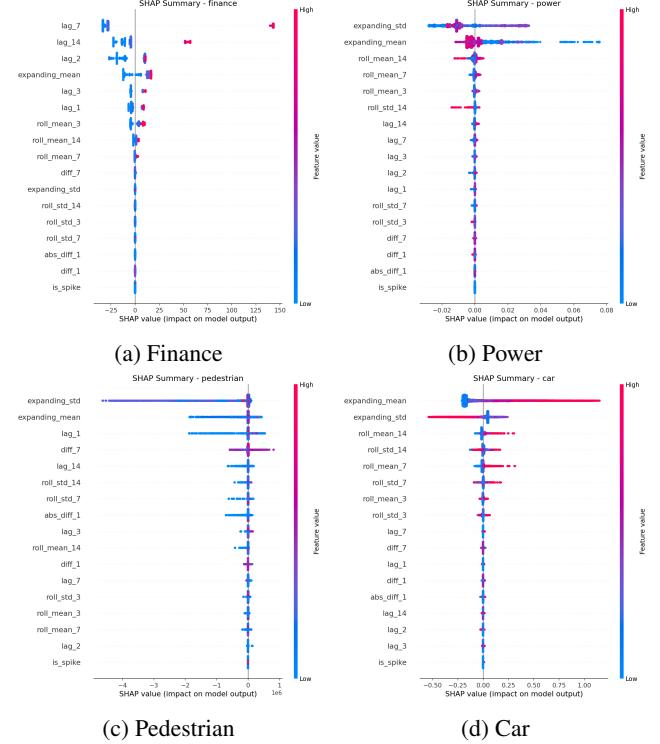


Figure B.2: SHAP values for Chronos Surrogate model across different domains.

Explains discrepancy: decent SMAPE (mean-relative) but catastrophic MASE (vs naive).

For **Power**, lag-30 (30-minute intervals) emerged as most critical, likely reflecting meeting schedules in the observed building. Hourly averages ranked second, capturing behavioral routines. The feature importance profile mirrors human activity patterns: arrivals, lunch breaks, and departures create predictable power demand fluctuations. With regards to **Pedestrian**, lag-168 (weekly seasonality) and Lag-24 (daily cycle) dominated, reflecting strong periodicity in foot traffic. Surprisingly, a binary weekend indicator showed minimal impact - likely because weekly lags already encode this structural information. Sensor location heterogeneity (business vs. leisure zones) may explain why simple temporal markers underperformed compared to lagged values. In the final domain, **Car**, the most important feature was whether the previous observation was zero, highlighting the dataset's extreme sparsity. Engineered features outperformed raw lags, demonstrating the need for domain-specific representations in sparse, spike-prone environments. All models failed to achieve satisfactory performance (MASE > 1.0), underscoring the unique challenges of zero-inflated time-series.

C LIME Analysis

Although the Segment LIME explanations provided valuable insights into which portions of the historical data most influenced specific forecasts, they also have notable limitations. The results are highly sensitive to how the time-series

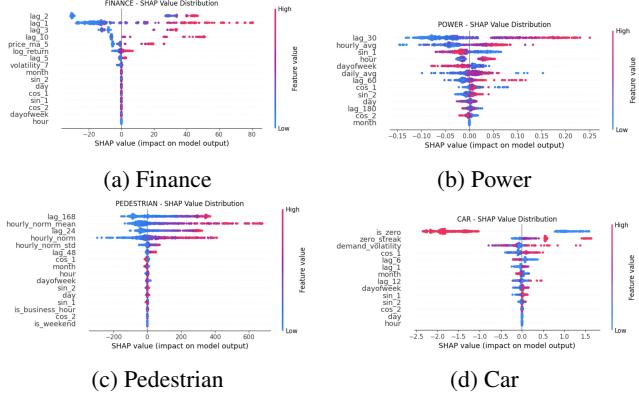


Figure B.3: SHAP values for Gradient Boosting model across different domains.

is segmented, with different segment lengths or boundaries often producing substantially different importance patterns. The sensitivity to the time-series segmentation can be seen comparing the importances from Figures 5 and C.5 which show the method applied to finance with 20 and 10 segments respectively. This sensitivity raises concerns about the stability and reproducibility of the explanations. Furthermore, although the highlighted segments offer an intriguing view of the model’s local dependencies, their direct usefulness for decision-making or for building a clear mental model of the underlying forecasting process remains uncertain. In practice, these explanations should be interpreted as exploratory aids rather than definitive accounts of model behavior.

Similar concerns are raised when looking at the effect of perturbation method. In the car domain, we tested zero perturbation (C.7) and inverse-max perturbation (C.6), the former replacing targeted segments with zeros and the latter replacing targeted segments with the inverse of the maximum value in the time-series. The car domain is unique in the sparseness of part sales. This property of the data caused most explanations show zero feature importance for all segments when using zero perturbation. This highlights a constraint of perturbing time-series data in that the strategy that is chosen to remove information can be dependent on several factors which include the model used and the time-series data itself.

In pedestrian, figure C.4 shows a similar distribution of feature importance, but has a noticeable bias of negative feature importance for earlier segments and higher feature importance for later ones. The explanation also shows that the majority of segments that fully capture high-traffic days with peaks at rush hours significantly drive the forecast down.

It should be noted that explanations for ARIMA on the pedestrian and power time-series were omitted due to the infeasibility of generating them. The implementation used in this paper refits the ARIMA model for each step in the forecast, causing it to scale poorly on both input length and forecast horizon, making these explanations infeasible for practical use.

Domain	Metric	Chronos	Chronos-S
Finance	MASE	5.48 ± 5.44	5.60 ± 5.34
	sMAPE	5.30 ± 5.15	5.40 ± 5.05
Power	MASE	2.33 ± 0.00	1.29 ± 0.00
	sMAPE	$8.84\% \pm 0.00\%$	$5.02 \pm 0.00\%$
Pedestrian	MASE	$1.47 \pm 0.67^*$	0.71 ± 0.24
	sMAPE	$90.98 \pm 44.77^*$	38.88 ± 13.29
Car Parts	MASE	∞	0.79 ± 1.32
	sMAPE	$175.03 \pm 37.56^*$	190.49 ± 16.30

Table B.1: Forecasting performance comparison. MASE and sMAPE are reported as (mean \pm std. dev.). sMAPE is reported as percentage (%). Chronos Surrogate \rightarrow Chronos-S. Chronos-S performs comparably to or better than Chronos without domain specific feature engineering. * Chronos has already seen these datasets in training and our results may differ from the authors’ due to different experimental setups.

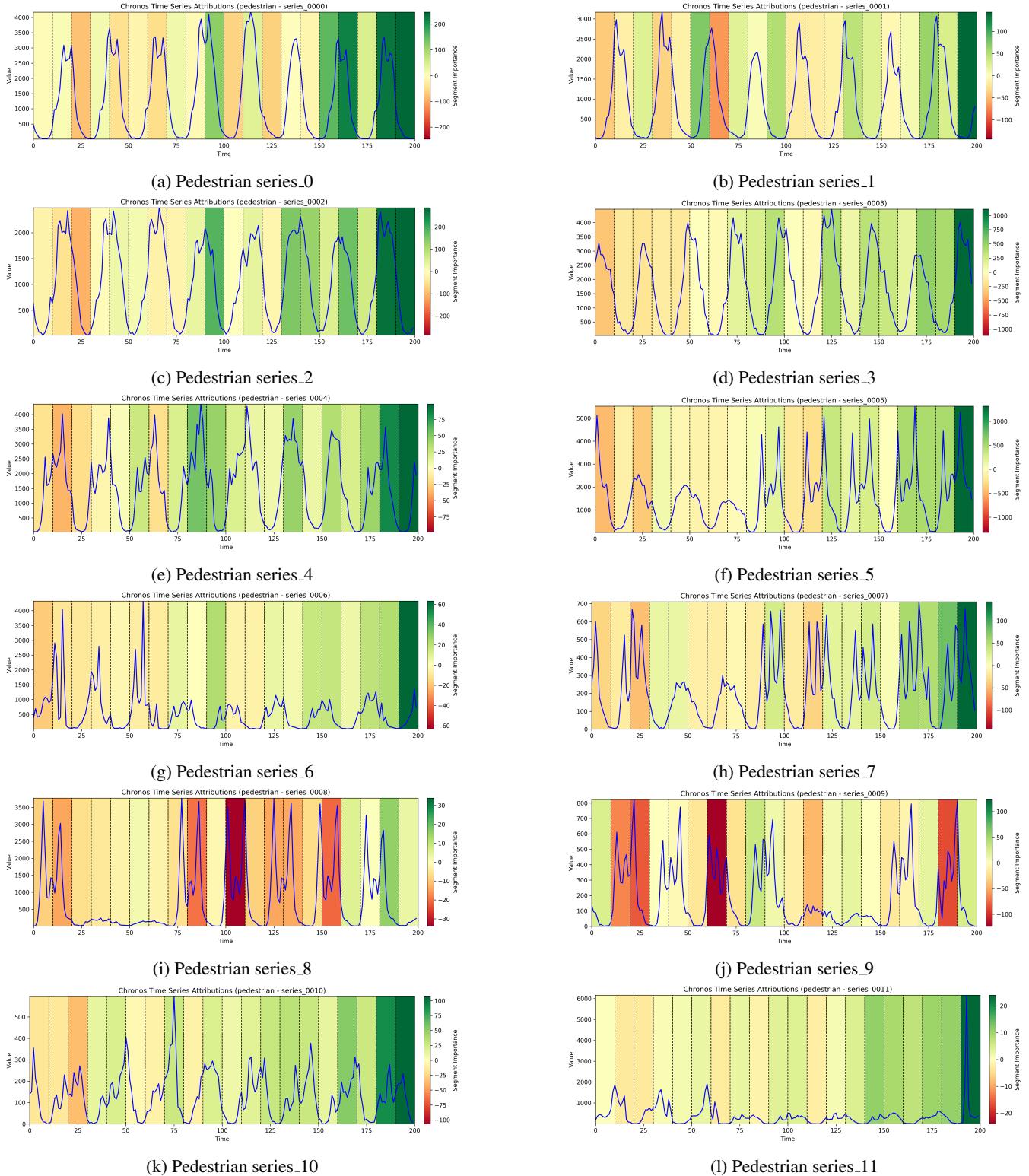


Figure C.4: LIME plots for Chronos model in the pedestrian domain using 20 uniform segments, zero replacement, and 100 samples on the last 200 points of the input time-series. Figures show the distribution of feature importance for each time-series, revealing a bias for negative feature importance in earlier segments and positive feature importance in later segments.

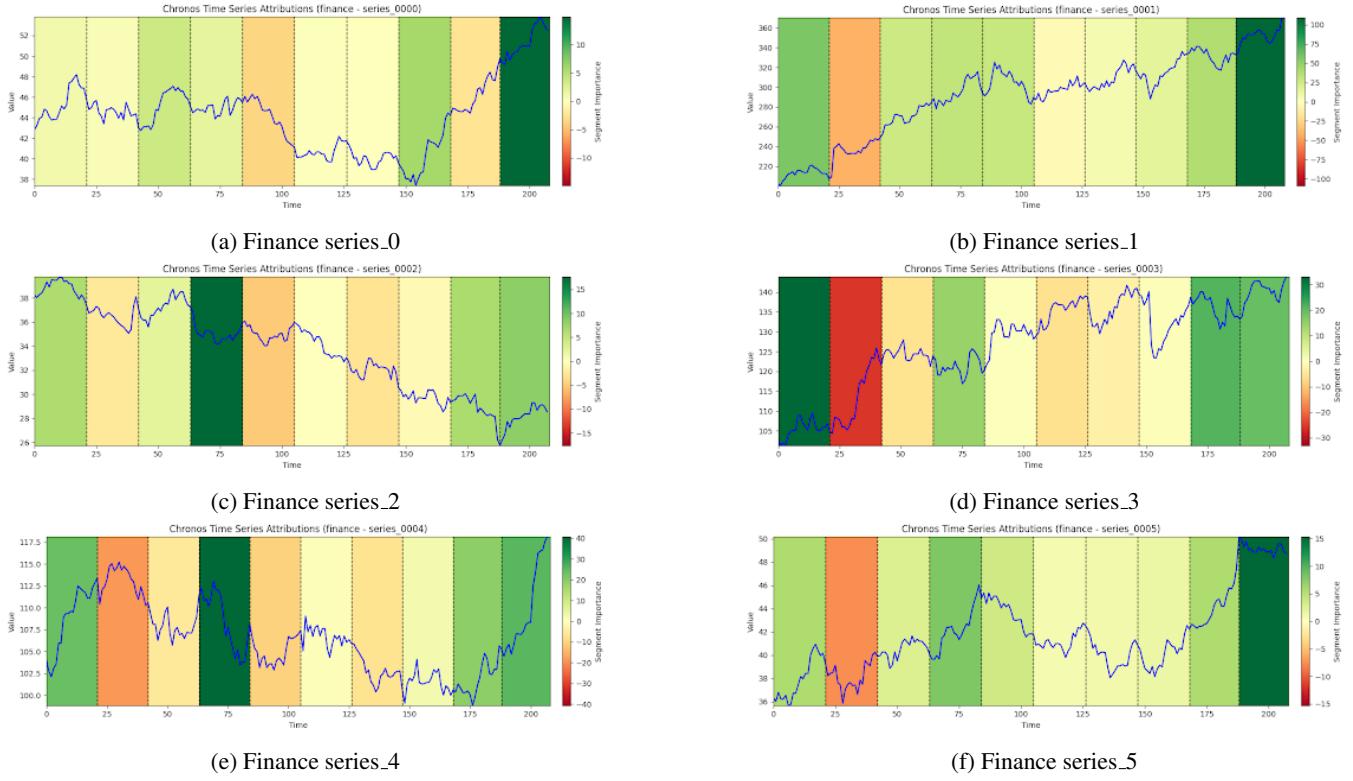


Figure C.5: LIME plots for Chronos model in finance using 10 uniform segments, zero replacement, and 200 samples.

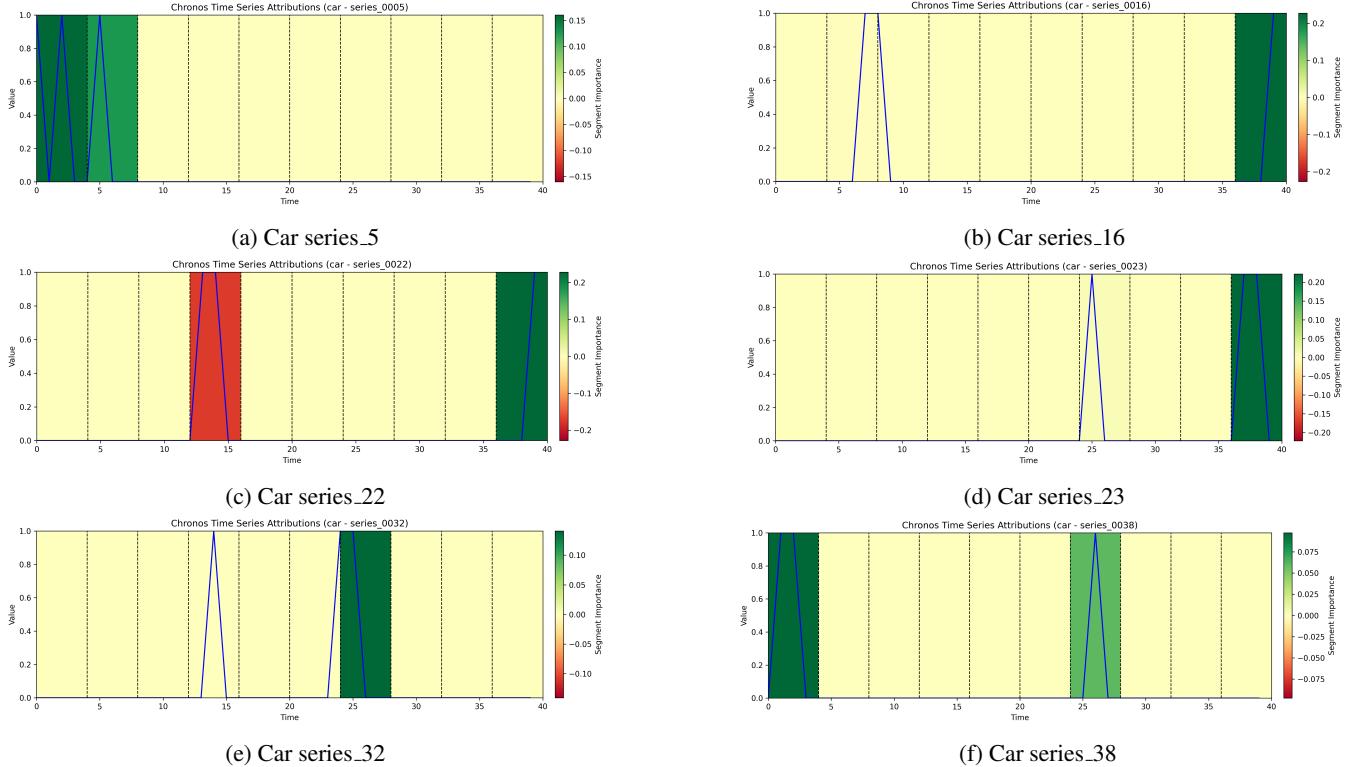


Figure C.6: LIME plots for Chronos model in the car domain using 10 uniform segments, zero replacement, and 100 samples.

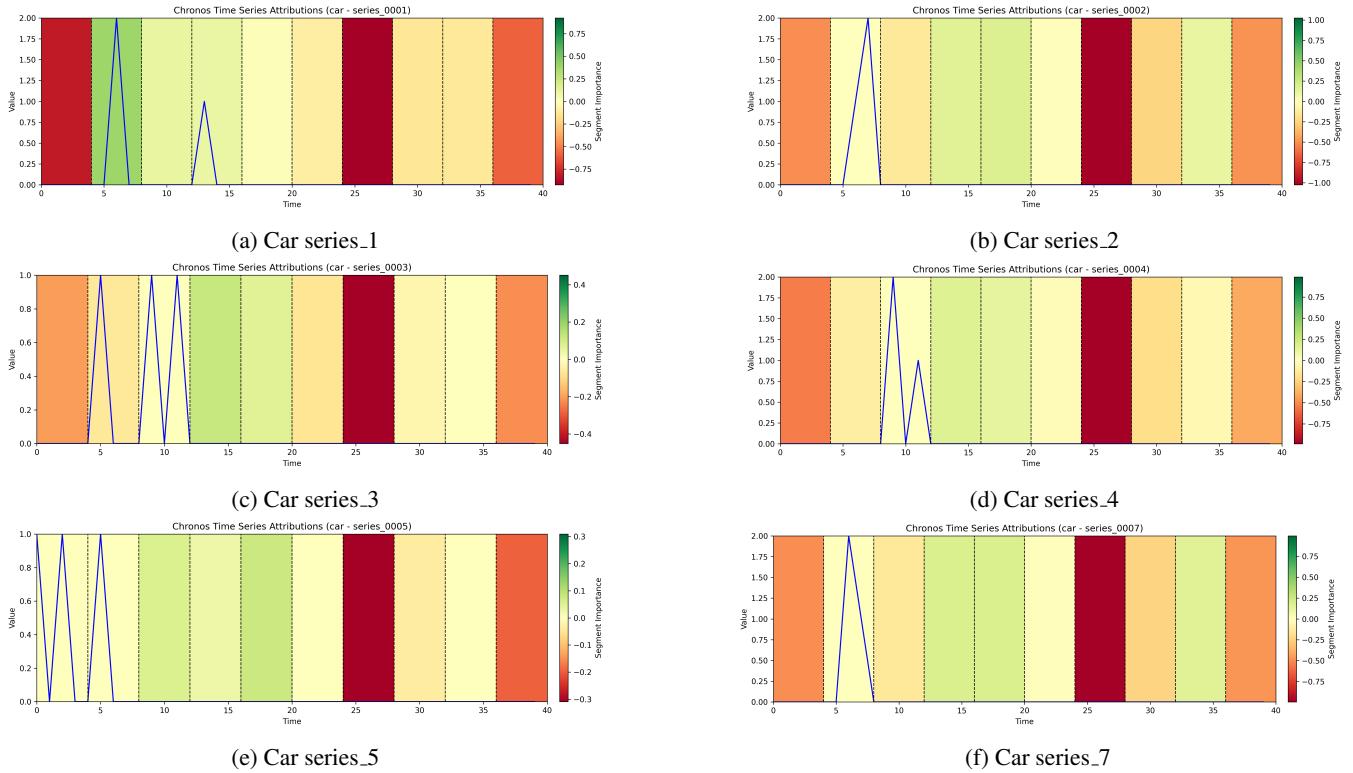


Figure C.7: LIME plots for Chronos model in the car domain using 10 uniform segments, inverse-max replacement, and 100 samples.

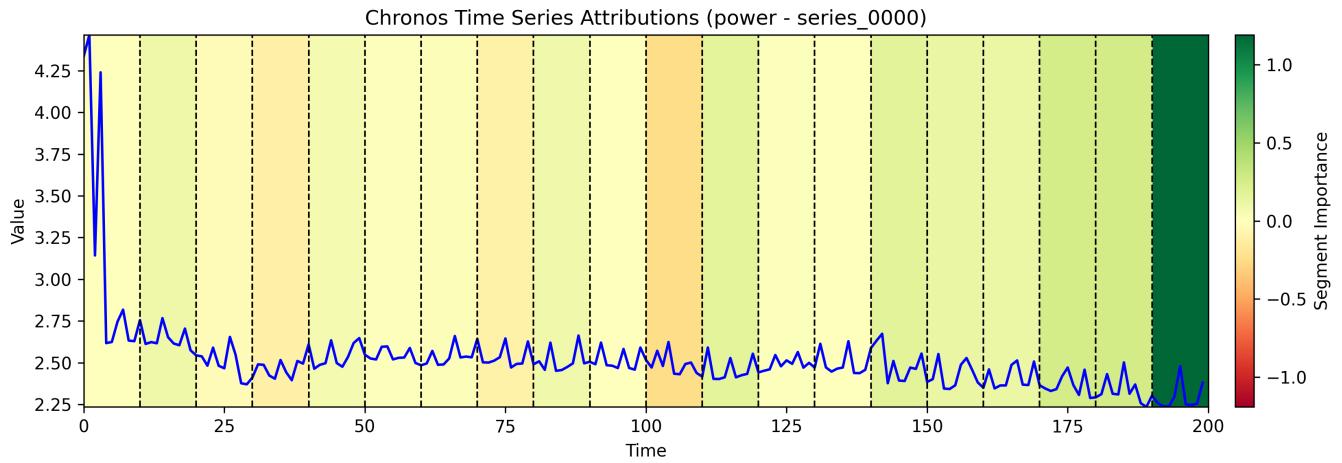


Figure C.8: LIME plot for Chronos model in power using 20 uniform segments, zero replacement, and 100 samples on last 200 points of the input time-series.