

XForecast: Evaluating Natural Language Explanations for Time Series Forecasting

Taha Aksu*, Chenghao Liu, Amrita Saha, Sarah Tan, Caiming Xiong, Doyen Sahoo

Salesforce AI Research,

{iaksu, chenghao.liu, amrita.saha, sarah.tan, cxiong, dsahoo}@salesforce.com

Abstract

Time series forecasting aids decision-making, especially for stakeholders who rely on accurate predictions, making it very important to understand and explain these models to ensure informed decisions. Traditional explainable AI (XAI) methods, which underline feature or temporal importance, often require expert knowledge. In contrast, natural language explanations (NLEs) are more accessible to laypeople. However, evaluating forecast NLEs is difficult due to the complex causal relationships in time series data. To address this, we introduce two new performance metrics based on simulatability, assessing how well a human surrogate can predict model forecasts using the explanations. Experiments show these metrics differentiate good from poor explanations and align with human judgments. Utilizing these metrics, we further evaluate the ability of state-of-the-art large language models (LLMs) to generate explanations for time series data, finding that numerical reasoning, rather than model size, is the main factor influencing explanation quality.

1 Introduction

Time series forecasting is essential in various fields: in healthcare, it assists in early triage assessment (Morid et al., 2021); in finance, it helps detect investment opportunities (Sezer et al., 2019; Leung and Zhao, 2020); in human resources, it aids workforce planning (Singh et al., 2024a); and in energy, it promotes efficient consumption (Deb et al., 2017). For decision-makers to rely on forecasts, trust and transparency are essential, necessitating explanations. Explainable AI (XAI) is a field dedicated to addressing this challenge for general machine learning applications.

XAI has also emerged in forecasting, where explanations typically use feature importance or saliency maps (Zhang et al., 2024; Pan et al., 2020;

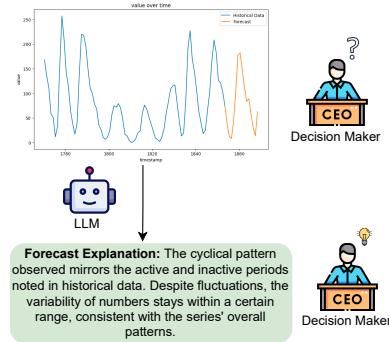


Figure 1: Example natural language explanation (NLE) for a time series forecast. While the raw forecast might be challenging for a layperson to interpret, the NLE provided by the LLM helps clarify the causal relationship.

Crabbe and van der Schaar, 2021; Raykar et al., 2023; Rajapaksha et al., 2021). However, these explanations require technical expertise, making them suitable for AI engineers rather than end users (Fok and Weld, 2023; Kaur et al., 2020; Lakkaraju et al., 2022; Mavrepis et al., 2024; Singh et al., 2024b; Du et al., 2018). In contrast, natural language explanations (NLEs) are more user-friendly and can explain predictions from black-box models to a broader audience (Lakkaraju et al., 2022; Mavrepis et al., 2024), see Figure 1 for an example.

Despite its usefulness, evaluating natural language explanations for forecasts is challenging due to the unclear causal relationship between the historical and forecast windows of time series data. Unlike structured data like tables, time series data requires further processing to uncover patterns (Sharma et al., 2021). This complicates distilling the technical depth of forecasting into concise NLEs and verifying their correctness.

To address this challenge, we propose two simulatability-based metrics: direct and synthetic simulatability. Direct simulatability measures how well a human can predict the black-box model's

*Taha Aksu is the corresponding author. Email: iaksu@salesforce.com

outputs from explanations of the original time series. Synthetic simulability tests how well the explanation generalizes to new, generated time series, ensuring it builds a mental model of the forecaster rather than overfitting to specific cases. To automate evaluation, we use a large language model (LLM) as a human surrogate following prior work (Aher et al., 2022; Argyle et al., 2022; He et al., 2023; Gilardi et al., 2023). To address the lack of annotated datasets of forecast-explanation pairs, we design a baseline explainer inspired by Warner (1998c) and conduct experiments to demonstrate the effectiveness of our proposed metrics. Our contributions are summarized as follows:

- **New performance metrics:** We introduce two complementary simulability-based performance metrics. Sanity checks and qualitative examples demonstrate their ability to distinguish between good and poor explanations. A two-part human study shows strong agreement between these metrics and human evaluations (Cohen’s Kappa of 0.42 and 0.58), supporting their effectiveness in assessing explanations based on their usefulness to humans.
- **Evaluating SoTA LLMs:** We then use these metrics to rank explanations generated by various LLMs. Our findings indicate that while LLM size impacts results, numerical reasoning capability is more crucial.

2 Related Work

Natural Language Explanations Pre-LLM XAI techniques mainly used structural explanations like saliency maps or feature importances due to the difficulty of generating text explanations. However, end users prefer text-based explanations. For example, Lakkaraju et al. (2022) found that decision-makers such as doctors prefer natural language explanations, wanting to treat ML models as "another colleague." Similarly, Mavrepis et al. (2024) found that over 80% of participants preferred ChatGPT-refined XAI outputs in natural language. Du et al. (2018) noted that many interpretable ML explanations are based on researchers’ intuition rather than user needs, making them suitable for developers but not for lay users, and suggested conversational explanations as a more user-friendly alternative. Likewise, Singh et al. (2024b) argued that LLMs can replace early XAI model interfaces, like saliency maps, enabling users to make targeted queries towards LLM-generated explanations.

Evaluation of XAI Techniques Several techniques evaluate XAI methods, with plausibility and faithfulness being the most common. Plausibility measures how convincing the interpretation is to humans, while faithfulness assesses how accurately it reflects the model’s true reasoning (Jacovi and Goldberg, 2020). However, these metrics alone are insufficient. For instance, Fok and Weld (2023) reviewed many studies and concluded that explanations are only useful if they help human decision-makers verify AI predictions. Kaur et al. (2020) found that data scientists often over-trust and misuse interpretability tools, raising concerns about the quality of explanations provided to executives. Another important metric is simulability, where humans must correctly simulate the model’s output based on the explanation and input (Doshi-Velez and Kim, 2017; Chandrasekaran et al., 2018; Hase and Bansal, 2020). Chen et al. (2023) applied simulability to explain LLMs answers to text questions. They describe simulability as a specific form of faithfulness, requiring human judgment rather than arbitrary black-box models. Since not every explanation is interpretable by humans, faithfulness does not imply simulability. This concept guided the design of our proposed metrics as we adapted similar concepts to the time series modality.

XAI for Time Series Forecasting Various techniques are employed to explain forecast predictions. Shapley values are commonly used to explain the causal relationship between a model’s features and its output. Zhang et al. (2024) proposed a general XAI approach based on Shapley values that explores explanations in the temporal dimension and feeds these back to the forecasting model to improve performance. Raykar et al. (2023) introduced TsSHAP, which uses the TreeSHAP algorithm on a surrogate model to explain forecasts in terms of user-defined interpretable features. Building on LIME (Ribeiro et al., 2016), Rajapaksha et al. (2021) provide local model-agnostic interpretations by training simple surrogate models on samples within a neighborhood of the time series being explained. Pan et al. (2020) and Crabbe and van der Schaar (2021) convert multivariate time series into images to find saliency maps reflecting the importance of each temporal range across different features. While saliency maps and interpretable features are useful for domain experts or AI engineers, they are less intuitive for stakeholders.

Algorithm 1 Evaluate Direct Simulability

Require: H : Time series data
Require: FM : Forecasting model
Require: $Explainer$: Forecast explainer
Require: $H.S.$: Human surrogate to simulate the forecast
Ensure: DS : Direct simulability score

- 1: $F \leftarrow FM(H)$ \triangleright Generate forecast
- 2: $NLE \leftarrow Explainer(H, F)$ \triangleright Generate explanation
- 3: $F' \leftarrow H.S.(H, NLE)$ \triangleright Human surrogate simulates forecast using NLE
- 4: $DS \leftarrow Distance(F, F')$ \triangleright Calculate the distance between F and F'

We measure the distance between the actual and predicted forecasts—the better the explanation, the smaller the distance.

3.3 Synthetic Simulability

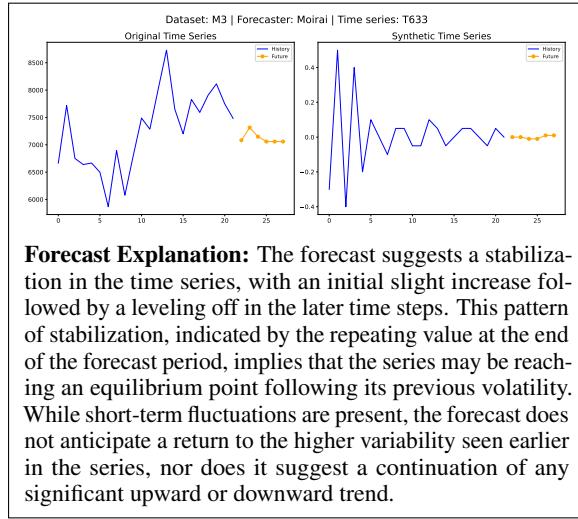


Figure 3: The original and synthetic time series history and forecast pairs generated using the Figure 2 pipeline (right) both align with the explanation, though they differ in scale and noise. The explanation is generated by the explainer using the original history and forecast.

A limitation of direct simulability is that it only evaluates how well an explanation simulates the original time series. However, a useful explanation should help form a mental model that generalizes beyond a single series. To address this, we introduce synthetic simulability, which assesses how well the explanation generalizes to new time series following the same reasoning.

Unlike Chen et al. (2023), who use the same LLM to generate questions and explanations, our task is to generate time series data with explanations from a separate system. While their input is text-based, we generate numerical sequences, which is challenging for an LLM. To address this, we follow Merrill et al. (2024) by using GPT-4 to

Algorithm 2 Evaluate Synthetic Simulability

Require: H : Time series data
Require: FM : Forecasting model
Require: $Explainer$: Forecast explainer
Require: $H.S.$: Human surrogate to simulate the forecast
Require: LLM : Large language model to generate new time series NLE
Require: PI : Python interpreter
Ensure: IS : Synthetic simulability score

- 1: $F \leftarrow FM(H)$ \triangleright Generate forecast
- 2: $NLE \leftarrow Explainer(H, F)$ \triangleright Generate explanation
- 3: $PF \leftarrow LLM(NLE)$ \triangleright Generate python function
- 4: $H_{new} \leftarrow PI(PF)$ \triangleright Run python function to generate new time series
- 5: $F_{new} \leftarrow FM(H_{new})$ \triangleright Generate new forecast
- 6: $F'_{new} \leftarrow H.S.(H_{new}, NLE)$ \triangleright Human surrogate simulates new forecast using NLE
- 7: $IS \leftarrow Distance(F_{new}, F'_{new})$ \triangleright Calculate the distance between F_{new} and F'_{new}

generate code that simulates time series data based on the explanations. This approach allows us to create new time series aligned with the explanations for synthetic simulability evaluation.

Synthetic simulation pipeline (*cf.* right side of Figure 2, and Algorithm 2) starts with the black-box model generating a forecast and the explainer producing an explanation. GPT-4 then generates a new time series from this explanation using NL-to-code generation, where it creates a Python function to simulate the series (*cf.* Appendix B for detailed prompts and examples). The black-box model forecasts this new series, and the human surrogate predicts the forecast using the original explanation. We compare the model’s forecast with the surrogate’s prediction, and the distance between them quantifies synthetic simulability: the smaller the distance, the better the explanation.

We present a sample synthetically generated time series in Figure 3. Both the original and synthetic series align well with the explanation—e.g., both forecasts show stabilization without trends, and both historical contexts display high volatility with short-term fluctuations. However, their scales and noise levels differ significantly, so an explanation overfitting the original series would not aid in predicting the synthetic forecast. Additional examples are available in Appendix D, Figure 7.

4 Baseline Explainer Design

To thoroughly test our evaluation metrics, we design a baseline explainer for generating NLEs for forecasts, as no existing baselines are available. For this baseline we refer to early work by Warner (1998c), which offers techniques for interpret-

ing time series data without requiring specialized knowledge. According to Warner (1998a,b) the key steps for explaining time series are: *i*) **Statistics:** Screen the data to assess distribution, outliers, and relevant characteristics. *ii*) **Trend:** Analyze linear trends to determine how much variance they account for. *iii*) **Seasonality:** Look for cyclic patterns in the data. *iv*) **Cycle Inconsistencies:** Describe changes in cycle irregularities, such as variations in peak amplitude over time. Following the steps listed above, our pipeline first extracts characteristics using statistical methods. We then iteratively prompt the LLM to generate a summary of the time series, the forecast, and the relationship between the two. For detailed prompts and specific examples, refer to Appendix C.

Given the length of time series data, we apply these steps to smaller segments and prompt the LLM to explain each separately, then aggregate them into a final explanation. Following Sharma et al. (2021), our pipeline first segments the time series based on slope changes. We then: *i*) Perform quantitative analysis on each segment, calculating and formatting trend, seasonality, mean, and standard deviation into a templated summary; *ii*) Concatenate the segment analyses and prompt the LLM to generate a comprehensive analysis of the full time series; *iii*) Provide the LLM with the historical data, black-box forecast, and comprehensive analysis to generate a short report interpreting the forecast. See *cf.* Figure 4 for a sample explanation generated by this pipeline.

5 Experiments and Analysis

5.1 Experimental setup

Datasets We collected time series data from three datasets in the Monash Repository (Godahewa et al., 2021): Tourism, M3, and M1. To ensure the effectiveness of our metrics, the backbone LLM must perform reasonably well in forecasting. We focused on yearly frequencies due to shorter sequence lengths, as LLM forecasting performance declines with longer sequences (Fons et al., 2024). As LLMs improve in time series reasoning, our metrics can be applied to any frequency. Exploration of other frequencies is left for future work.

Models To test if our evaluation metrics are forecasting method-agnostic, we selected diverse models, including statistical methods like auto ARIMA and auto ETS (Garza et al., 2022), deep learning models like DeepAR (Flunkert et al., 2017), and

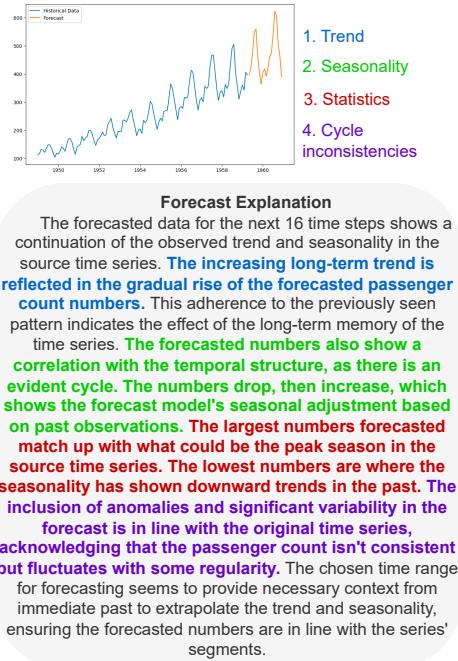


Figure 4: Sample forecasting explanation generated by our pipeline. The colored snippets refer to the 4 salient points by Warner (1998c) when explaining time series.

transformer-based models like Moirai (Woo et al., 2024) and PatchTST (Nie et al., 2022). These forecasters made predictions on the datasets, and we generated explanations using the baseline method from Section 4 with various SoTA LLMs, both open and closed source. The resulting dataset of time series, forecast, and explanation triplets was used to evaluate our metrics and compare LLMs' explanation capabilities.

For both metrics, we first run sanity checks with GPT-4 generated explanations, using three baselines: LLMTIME: a naive baseline that predicts the forecast without any explanation; LLMTIME_R: which uses the pipeline described in Section 4, but explains a random forecast; LLMTIME_M: which prompts LLMTIME to predict a constant value for all steps. We compare these baselines against LLMTIME_E which uses the correct forecast in the pipeline from Section 4.

Through sanity checks, we expect LLMTIME_E, using the correct explanation, to predict the forecast better than the other baselines. Afterward, we extend the experiments to other LLMs to compare how explanations from different models improve forecast prediction. Hyperparameter settings for

Model Size and Performance. Judging by the Llama2-70B results, model size alone does not guarantee high-quality explanations. For example, Vicuna-7b outperforms Llama2-70B on most dataset-forecaster pairs across both metrics, despite having ten times fewer parameters. This suggests that numerical reasoning may correlate with better time series reasoning, as Vicuna-7b has demonstrated stronger numerical reasoning compared to larger models (Zheng et al., 2023).

Explanations Across Forecaster Families: A key question is whether explanations behave similarly across model families. We use the direct simulability metric for comparison, as it uses the same time series to simulate each forecaster’s predictions. As shown in Table 2, PatchTST and Moirai have the largest error values for simulability, likely due to patch embeddings and our use of short time series sequences. When the context is shorter than the minimum patch size, these models underperform, affecting explanation quality. In contrast, statistical models show the smallest error, likely because their simpler behavior is easier to explain in natural language.

5.2.4 Human Study

We conduct a two-part human study to evaluate the effectiveness of our proposed metrics in distinguishing useful explanations, see Appendix F.2 for setup details. We sample 20 examples where both direct and synthetic simulability metrics agree on whether an explanation is "useful" or "not useful." An explanation is deemed useful if it improves time series forecast accuracy, and these examples are used in both parts of the study.

In the first part, participants replaced the human surrogate in Figure 2, and were asked to draw forecast for 20 time series, initially without any explanation, and then with explanations. We calculated the average improvement in the forecast for both useful and non-useful explanations. The results show that explanations deemed useful by our metrics improved accuracy by 5%, while non-useful explanations reduced it by -11%. This validates our metrics’ ability to identify helpful explanations, with a Kappa score of 0.42 showing moderate agreement between user-improving explanations and those classified as useful.

In the second part of the study, participants were shown the ground truth forecast and asked to assess the usefulness of the explanation. This evaluation

yielded a Kappa score of 0.58, indicating moderate agreement with our metrics. These results confirm that human judgment aligns well with our metrics, validating their reliability in distinguishing between useful and non-useful explanations.

6 Future Directions

Time-series and language foundation models
This study is a first step in bridging time-series and language foundation models. Future work could explore using NLEs to fine-tune models capable of processing both modalities.

Applications of performance metrics Our metrics can be applied beyond evaluation, such as in n-best sampling or as rewards for fine-tuning LLMs to generate better explanations, similar to Chen et al. (2024)’s work.

Other downstream time-series tasks While we focus on forecasting, future work could explore NLE generation and evaluation for other tasks like time-series classification and anomaly detection.

7 Conclusion

We propose two complementary metrics for evaluating time series forecasting explanations: *i*) direct simulability, which measures how well explanations help predict the black-box model’s forecast on the original time series, and *ii*) synthetic simulability, which evaluates how well the explanation generalizes to synthetic time series generated from the explanation. Through sanity check experiments, we demonstrate that both metrics can distinguish between good and poor explanations. We then use these metrics to evaluate various LLMs’ explanation performance. Our findings are two-fold: first, model size impacts explanation performance, but numerical reasoning capability is more critical; second, explanations for statistical model forecasters provide more insight than those for deep learning and transformer-based forecasters. A human study shows high agreement between our metrics and human annotators, supporting our premise that explanations must help humans understand the model’s behavior. Finally, qualitative analysis shows that high-quality explanations improve LLM predictions, validating our metrics’ effectiveness. We hope this study encourages further exploration of natural language explanation pipelines for time series forecasting.

- Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. Association for Computational Linguistics.
- Xingwei He, Zheng-Wen Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2023. Annollm: Making large language models to be better crowdsourced annotators. *ArXiv*, abs/2303.16854.
- Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.
- Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. 2020. Interpreting interpretability: Understanding data scientists’ use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, page 1–14, New York, NY, USA. Association for Computing Machinery.
- Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chen-hao Tan, and Sameer Singh. 2022. Rethinking explainability as a dialogue: A practitioner’s perspective. *ArXiv*, abs/2202.01875.
- Tim Leung and Theodore Zhao. 2020. Financial time series analysis and forecasting with hht feature generation and machine learning. *EngRN: Signal Processing (Topic)*.
- Philip Mavrepis, Georgios Makridis, Georgios Fatouros, Vasileios Koukos, Maria Margarita Separdani, and Dimosthenis Kyriazis. 2024. Xai for all: Can large language models simplify explainable ai? *ArXiv*, abs/2401.13110.
- Mike A. Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. 2024. Language models still struggle to zero-shot reason about time series. *ArXiv*, abs/2404.11757.
- Mohammad Amin Morid, Olivia R. Liu Sheng, and Josef A. Dunbar. 2021. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14:1 – 29.
- Shikai Qiu Nate Gruver, Marc Finzi and Andrew Gordon Wilson. 2023. Large Language Models Are Zero Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*.
- Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *ArXiv*, abs/2211.14730.
- Qingyi Pan, Wenbo Hu, and Jun Zhu. 2020. Series saliency: Temporal interpretation for multivariate time series forecasting. *ArXiv*, abs/2012.09324.
- Dilini Sewwandi Rajapaksha, Christoph Bergmeir, and Rob J Hyndman. 2021. Lomef: A framework to produce local explanations for global model time series forecasts. *ArXiv*, abs/2111.07001.
- Vikas Chandrakant Raykar, Arindam Jati, Sumanta Mukherjee, Nupur Aggarwal, Kanthi Kiran Sarpatwar, Giridhar Ganapavarapu, and Roman Vaculín. 2023. Tsshap: Robust model agnostic feature-based explainability for time series forecasting. *ArXiv*, abs/2303.12316.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “why should i trust you?”: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2019. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *ArXiv*, abs/1911.13288.
- Mandar Sharma, John S. Brownstein, and Naren Ramakrishnan. 2021. T3: Domain-agnostic neural time-series narration. *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1324–1329.
- Brijesh Singh, Rupali Khaire, and Sashikala V. 2024a. Comparative analysis of time series forecasting methods in workforce planning using predictive analytics. In *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, pages 1–4.
- Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024b. Rethinking interpretability in the era of large language models. *ArXiv*, abs/2402.01761.
- Rebecca M Warner. 1998a. *Chapter 1: [Research Questions]*, pages 4–7. In (Warner, 1998c).
- Rebecca M Warner. 1998b. *Chapter 7: [Summary of Univariate Time Series Data]*, pages 100–102. In (Warner, 1998c).
- Rebecca M Warner. 1998c. *Spectral analysis of time-series data*. Spectral analysis of time-series data. Guilford Press, New York, NY, US.
- Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*.
- Yuyi Zhang, Qiushi Sun, Dongfang Qi, Jing Liu, Ruimin Ma, and Ovanes Petrosian. 2024. Shapetime: A general xai approach for explainable time series forecasting. In *Intelligent Systems and Applications*, pages 659–673, Cham. Springer Nature Switzerland.

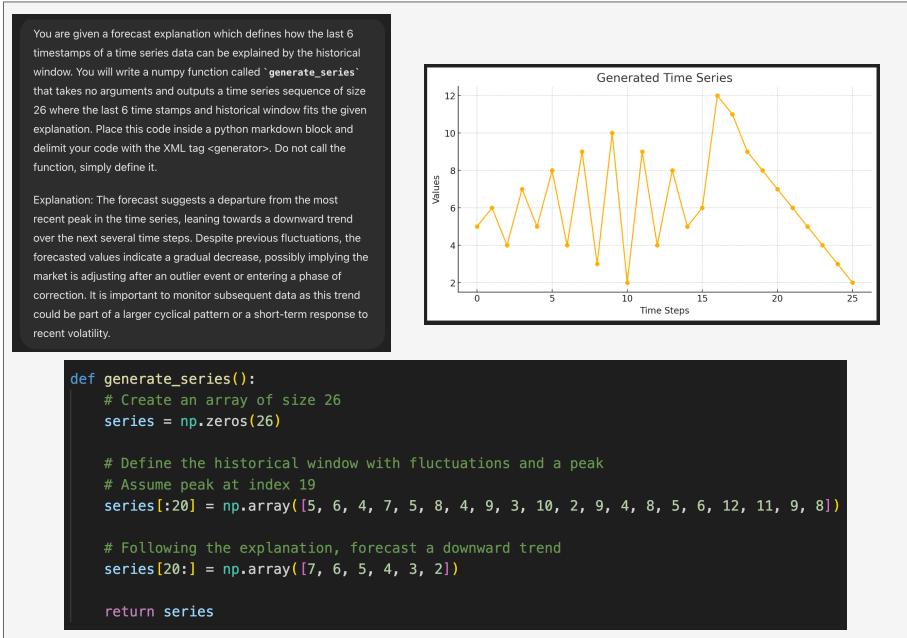


Figure 6: Sample time series generation through code as an intermediary using GPT-4. Using code as an intermediary gives the LLM higher control, allowing it to make changes that directly reflect the explanation.

You are given a paragraph that explains the reasoning behind forecasting result of some time series. Can you change it such that it is a recommendation to another user who needs to do forecast on the same time series. Try to keep the recommendations short up to two or three sentences.

Here is the paragraph:
{forecast_explanation}

Table 6: The prompt used to generate the forecast tip from the forecast explanation.

C Explanation Generation Pipeline and Prompts

In this section, we provide a detailed view of the tools and prompts used to design our baseline in Section 4. The first step in our explanation generation process is segmentation, following the method from [Sharma et al. \(2021\)](#), which separates the time series by fitting slopes to each segment. After segmentation, we calculate statistical data for each segment and summarize each segment using a templated string. We then concatenate all segment summaries and prompt the language model to generate a summary for all segments, referred to as {segment_analysis}, cf. Table 8. Next,

You are given a forecast explanation which defines how the last {forecast_horizon} timestamps of a time series data can be explained by the historical window. You will write a numpy function called ‘generate_series’ that takes no arguments and outputs a time series sequence of size {timeseries_size} where the last {forecast_horizon} time stamps and historical window fits the given explanation. Place this code inside a python markdown block and delimit your code with the XML tag <generator>. Do not call the function, simply define it.

Explanation: {forecast_explanation}

Table 7: The prompt used to generate the forecast tip from the forecast explanation.

You are a helpful assistant who is expert in understanding time series data.

You were given some time series data and used an external tool to find different segments in the data along with their slopes and mean and std values. Try to understand the segments and their characteristics and generate a brief analysis of the time series' segments such as seasonality, cycles and overall trend.

Here is an individual analysis of each segment: There are $\{N\}$ segments in the time series

1. Segment $\{k\}$ starts at index $\{\text{start_idx}\}$, ends at index $\{\text{end_idx}\}$. The mean is $\{\text{mean}\}$ the std is $\{\text{std}\}$ and the slope in this segment is $\{\text{trend}\}$. It repeats itself every $\{\text{seasonality}\}$ predictions.
- 2...

Based on the above information, generate an analysis with few sentences of the time series' segments with information such as seasonality, cycles and overall trend.

Table 8: The prompt used to aggregate templated segment explanations into final segment analysis.

we provide the LLM with the full time series data along with the $\{\text{segment_analysis}\}$ generated in the previous step and ask it to output a full analysis of the historical context (*cf.* Table 9). Finally, using the prompt depicted in Table 10 we provide the LLM with the time series data, the summary analysis from the earlier step, the forecast from the black-box forecaster, and a preliminary analysis of the forecast using (Zhang et al., 2024), which highlights the most important indices used for forecasting. The output of this final prompt is our time series forecasting explanation. For a sample of such an explanation, see Figure 4.

D Synthetically Generated Examples

The synthetic generation process ensures diversity by requiring the large language model (LLM) to create new time series that adhere to the same explanation, without revealing any information about the original series. This approach forces the LLM to generate time series that capture the high-level properties described in the explanation—such as trends, seasonality, or other structural patterns—while differing in aspects like scale, noise, or variability. By doing so, we can assess whether the explanation generalizes broadly to de-

You are a helpful assistant who is expert in understanding time series data. You are provided with the full length of time series data in comma separated format, along with some finegranular analysis of the time series data. Then you are asked to generate a short analysis report of the time series data. The report should help the user to forecast the next steps in time series data. Keep the analysis short and to the point. Avoid being redundant and don't suggest that further analysis is needed.

Here is the time series data in comma separated format:

```
{time_series_data}
```

Here is the analysis for all segments:

```
{segment_analysis}
```

Generate a short analysis with 2-3 sentences that explain the data in general terms and give hints for the forecaster.

Table 9: The prompt used to generate the final summary of the historical time series data.

scribe the forecaster's behavior or if it is overfitted to the specific values of the original time series. As shown in Figure 7, the original and synthetically generated series may differ in their specific data points, but they still follow the same reasoning laid out by the explanation. This diversity is crucial for testing the robustness of the explanation across different yet structurally similar time series, ensuring that it reflects the model's broader decision-making patterns rather than being overly tailored to one instance.

E Additional Qualitative Examples

Figure 8a shows an example explanation for a time series in the M1 dataset using the Auto ARIMA forecaster. The naive LLMTTime forecast, without an explanation, naturally assumes a continuation of the decreasing trend. However, ARIMA models assume that the data is stationary, leading to a prediction of a flattened line. The figure clearly shows that LLMTTime with the explanation does a much better job of predicting the main forecast. It is important to note that, as in earlier XAI work, we are not focused on the accuracy of the prediction itself but on whether the explanation is faithful to the main forecaster.

For Figure 8b the generated time series depicts a cyclical plot with an overall growing trend as

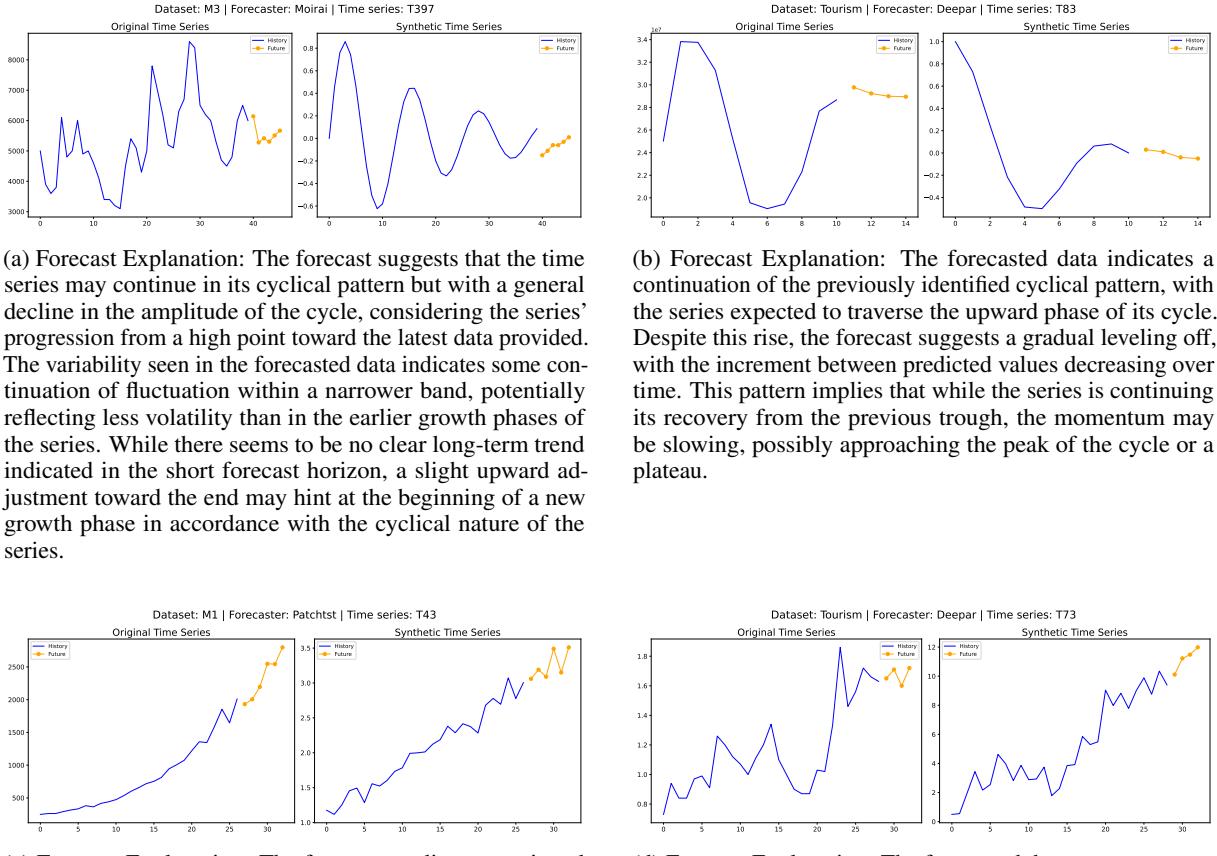
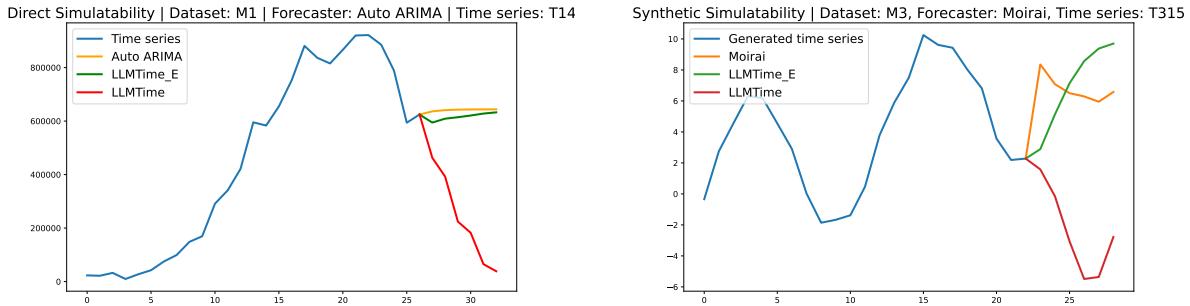


Figure 7: Synthetically generated time series alongside their original counterparts and corresponding explanations across different datasets and forecasting models.



(a) **Forecast Explanation:** The forecasted data suggests a stabilization in the time series following the recent sharp decline... the rather flat trajectory in the forecast points to a period of low variability, providing no strong signs of an immediate return to the previous high growth rates. Metric results: LLMTIME = 1.04, LLMTIME_E = 0.04.

(b) **Forecast Explanation:** The forecast suggests the time series will continue to display its cyclical nature with variability evident as it includes both rises and dips, indicating ongoing fluctuations within an overall growing trend... Metric results: LLMTIME = 1.89, LLMTIME_E = 0.42.

Figure 8: More qualitative examples with explanations and their respective effect on forecasting. Figure 8a shows an example extracted from simulability experiments, whereas Figure 8b is from synthetic simulability experiments.

You are a helpful assistant who is expert in explaining forecasts of time series data. You are provided with the full length of time series data in comma separated format, along with some finegranular analysis of the time series data. You are also provided with the estimated forecast of the data for the next {forecast_horizon} time steps.

Then you are asked to generate a short 2-3 sentences long interpretation of the forecasted data. Explain the forecasted data in terms of the time series' temporal structure, variability, long-term memory, trend, and seasonal pattern. Do not use specific data points or numbers in your analysis. Instead, focus on the general trends and patterns in the data.

Here is the time series in comma separated format:

{time_series_data}

Here is the analysis of the data:

{time_series_analysis}

Here is the forecasted data for the next {forecast_horizon} time steps:

{forecasted_data}

Here is a preanalysis of the forecast:

{forecast_panalysis}

Generate a short analysis reporting interpretation of the forecasted data with 2-3 sentences.

Table 10: The prompt used to generate the final explanation depicting the interaction between the historical time series data and the forecast.

mentioned in the explanation. Regarding the forecasting, it is possible to see that LLMTIME forecasting with the explanation again aligns better with the actual forecast compared to the no-explanation generation, which assumes a downward trend following the past few time steps.

F Experimental Setup Details and Additional Results

F.1 Hyper parameter setting for LLMs

We use five different LLMs in our experiments, with the only closed-source model being GPT-4 (specifically the ‘gpt-4-1106-preview’ model available via API). For GPT-4, we set the *temperature* to 1.0 following [Nate Gruver and Wilson \(2023\)](#). For all open-source LLMs (Llama3, Llama2, Mistral, and Vicuna), we use the following settings: *temperature* of 0.9, *top_p* of 0.9, and *rep_penalty* of 1.1. Inference with the GPT-4 model is done using the official OpenAI API, whereas all open-source models are run on 4 Nvidia A100 GPUs.

F.2 Human study instructions

Prior to sending each user the instructions we get their consent on how the data will be used through a different form. All annotators are daily english speakers.

[Figure 9](#) illustrates the setup for part 1 of the human study. Participants were instructed to generate forecasts for all 20 time series included in the study. For each series, they first drew their forecast without access to any explanation, and then repeated the task after being provided with the explanation.

Figure 10 shows the human study part 2 setup with three key components. First, detailed instructions were provided to participants, guiding them on how to evaluate the explanations for their usefulness in understanding the forecasts. Second, two sample annotations illustrated ‘useful’ and ‘not useful’ explanations along with their reasoning. These examples were included to help participants understand the evaluation criteria but were not part of the actual questionnaire. Third, is a sample pair where a time series forecast explanation and the corresponding forecast provided. The users were expected to label 20 such pairs either as ‘useful’ or ‘not useful’.

F.3 Additional Results

The main paper reported results using the sMAPE distance metric between forecasts. In this section, we share the full results, which also include NMAE and NRMSE metrics, *cf.* Tables 11 to 14.

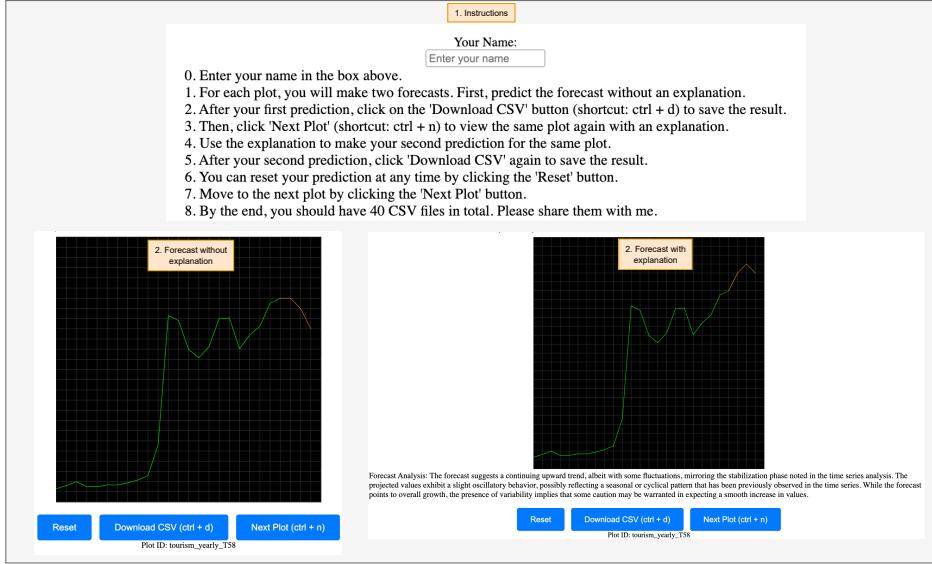


Figure 9: Overview of the human study part 1 setup. (1) Instructions provided to participants. (2) Participants are first asked to draw a forecast without explanation. (3) They then draw the forecast for the same time series using the explanation. This is repeated for all 20 time series that we have included in the human study.

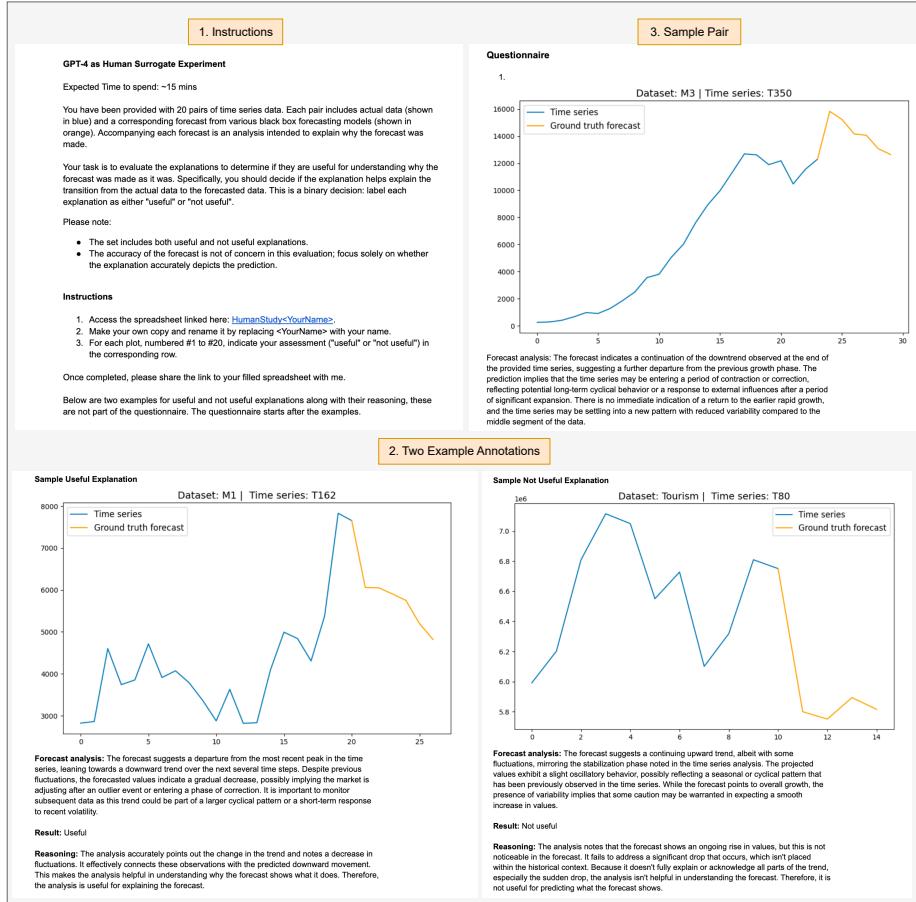


Figure 10: Overview of the human study part 2 setup. (1) Detailed instructions provided to participants. (2) Two example annotations illustrating ‘useful’ and ‘not useful’ explanations along with their reasoning. (3) A sample pair showing a time series forecast explanation and the corresponding forecast, used in the questionnaire.

