Time Series Forecasting using Lightweight LM on Synthetically-generated Signal

Avi Bleiweiss
avibleiweiss@bshalem.onmicrosoft.com
BShalem Research
Sunnyvale, California, USA

Abstract

Recent studies have unlocked strong sequential reasoning capacity in Large Language Models (LLMs), allowing them to substitute customized time-series models and serve as foundation models for complex temporal analysis. A line of research explored how to harness LLM power to effectively perform time-series tasks, including forecasting, anomaly detection, and fault cause localization. Approaches offered ranged from transforming time-series signals to linguistics embedding representation, generating synthetic time series, and using LLM prompting to convert natural language descriptions into numerical time-series predictions. In this paper, we employed the Chronos-T5 foundation model, a pretrained time-series forecasting model that leverages LM architectures to predict future data points through sampling. We evaluated the most parameter-compact Chronos-T5 variant on the NL2TS-675 benchmark that spans six diverse disciplines, and provide comprehensive analysis and forecasting visualizations across domains. Our under-resourced framework achieved a dataset-level MAE score comparable to the most recently published state-of-the-art, when using a prominent LM. The key novelty of our work lies in automatically aligning a time series with a domain at runtime using temporal-sequence clustering.

Keywords

Time Series, Forecasting, Large Language Model (LLM), K-Means Clustering

KDD,

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Avi Bleiweiss. 2026. Time Series Forecasting using Lightweight LM on Synthetically-generated Signal. In *Proceedings of (KDD).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Time-series forecasting is the task of predicting future values based solely on past observations. The task is an essential component of decision-making widely applied across diverse domains, including finance, transportation, energy, atmospheric, and healthcare, among others [12]. Recently, researchers noted a key observation that both LLMs and time series forecasting aim to decode sequential patterns to predict future events. This alignment allows to treat time series data as a language modeled by off-the-shelf transformer architectures. By encoding time series as a string of numerical digits, time series forecasting can be framed as next-token prediction in text, thus unlocking the use of powerful pretrained models and probabilistic capacities, such as likelihood evaluation and sampling [3]. Rather than reprogramming the LLM backbone to further augment the model reasoning about time-series patterns, Jin et al. [4] used prompt engineering to enrich the input time series with additional context and provide task instructions in the modality that better suited natural language representation. Large foundation models are therefore reduced to effective few-shot and zero-shot time-series learners when integrated through this reprogramming approach, outperforming state-of-the-art specialized forecasting models. Seeking a more generalized and uniformed framework, Zhou et al. [13] proposed a Frozen Pretrained Transformer (FPT) with a BERT encoder backbone [6] to perform temporal sequence evaluation through finetuning on all major types of tasks involving time-series forecasting.

Amazon had recently introduced Chronos [2], a family of pretrained deep learning models designed for time-series forecasting. Chronos models were trained on a large corpus of publicly available time series data after being transformed into a sequence of tokens via scaling and quantization. A language model is then trained on these tokens using cross-entropy loss. Chronos has been widely adopted since its

Table 1: NL2TS-675 data split distributions across domains.

	Finance	Healthcare	Weather	IoT	Technology	Retail	Overall
Train	121	70	53	66	66	96	472
Dev	27	9	20	13	11	21	101
Test	32	11	17	11	13	18	102

inception and considered a strong contender when it comes to zero-shot forecasting capacities, leveraging time-series data from diverse domains to improve accuracy on unseen prediction tasks. Chronos has shown to outperform traditional forecasting pipelines like ARIMA and Prophet, mainly in scenarios requiring longer prediction horizons. In our paper, we used the Chronos model trained on the Text-To-Text Transfer Transformer (T5) architecture [8]. We explored the chronos-t5-tiny variant of the Chronos-T5 foundation model, [1] which renders a scant 8M parameters and is optimized for efficient inference and suits resource-constraint settings.

Contribution. Our contributions include: 1) We propose a runtime temporal-sequence clustering that automatically assigns a domain to a time series. Rather than at dataset creation time that incurs both computation and storage cost, this approach is more scalable and effective for augmenting the dataset. 2) We provide extensive analysis of time-series forecasting using a less capable LM through both numerical and visualization cues. 3) Using a small LM, our experimental results are further contrasted with a large LM and prove almost an identical accuracy score for overall forecasting.

2 Related Work

The rapid development of LLMs has led to the creation of models that can perform a wide range of tasks, and begun to inspire new avenues in time-series research [5]. Building on this trend, recent works have explored time-series reasoning with LLMs by framing tasks as natural-language problems, including time-series understanding and question answering. Prompting domain-specific textual data are often encoded as prefix embeddings to enrich time series representations. Alternatively, methods leveraged zero-shot or few-shot capabilities of LLMs by directly prompting pretrained language models with text-converted time series, often yielding surprisingly strong performance in real-world scenarios of forecasting and classification tasks [7]. By pretraining on large multi-domain datasets, several time-series foundation models, such as Chronos [2] have achieved impressive zero-shot prediction capabilities on general purpose time-series benchmarks, eliminating the need for domain-specific training or finetuning.

The dearth of large and diverse time-series datasets had synthetic data generation emerge as a joint research line to improve generalized forecasting. Aiming to reduce the expensive process for training models from scratch, Rousseau et al. [10] introduces an efficient framework for generating synthetic time series using LLMs. The framework transforms both univariate and multivariate signals into tabular embeddings, which are then encoded into text and used to finetune any autoregressive LLM. By operating over compact tabular embeddings, they leverage foundation models with minimal finetuning cost. Across a broad range of datasets, their method outperforms existing generative models in similarity-based evaluations, and proven to enhance downstream forecasting performance.

In scenarios where histories are scarce or for rapid prototyping, Ram [9] proposed to convert freeform natural language descriptions into numerical time series predictions. They employed a multi-modal design comprising prompting an instruction-following LLM (Llama 3.1), while grounding domain and pattern specific predictions with lightweight numerical models. Across six domains they report an overall Mean Absolute Error (MAE) of 16.06. We evaluated our framework on the generated synthetic dataset NL2TS-675 they released for both finetuning and inference.

3 Dataset

NL2TS-675 [9] is a comprehensive benchmark dataset for converting natural language to time series predictions. The dataset consists of 675 description–series pairs across diverse domain and temporal patterns. Each instance in the dataset abides by the structure shown in Table 5 (Appendix A).

We follow with the interpretation of the eight fields that constitute an NL2TS-675 item: [2] 1) `uid:` A unique identifier of a dataset instance. 2) `domain:` NL2TS-675 traverses six domains, including finance, healthcare, weather, internet-of-things (iot), technology, and retail. 3) `text:` Presents a natural language description of the time series pattern. 4) `pattern:` A temporal sequence behavior (trend, seasonal, spike, plateau, and irregular). 5) `series:` A numerical time series as a list of floats that represents the historical timeline. 6) `freq:` A temporal frequency (hour, day, week, month, and

[1] https://huggingface.co/amazon/chronos-t5-tiny

[2] https://github.com/gokulsrinaths/NL2TS-675/tree/master

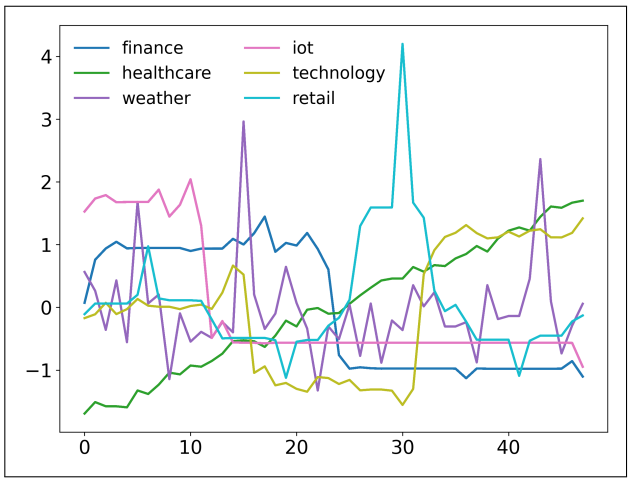


Figure 1: Visualization of learned time-series clustering centroids. Maximal sequence length is 48 time steps, and values are scaled, where $\mu = 0.$ and $\sigma = 1.$, for each time-series.

year). 7) `length`: Number of time steps in a temporal historical term (12, 24, or 48). 8) `split`: Data partition, one of train, dev, or test. We note that our dataset instance differs from the original by the added explicit `pattern` field and reducing the `uid` to a running numeric field.

In Table 1, we show the NL2TS-675 data split distributions across domains. Overall, the dataset utilizes a 70/15/15 partition percentage of train/dev/test sets, respectively. The finance domain is strikingly leading in terms of coverage and this corroborates the view of largest and most comprehensive available datasets across diverse economic topics. The remainder of the domains are reasonably balanced with a $\sigma \in (\text{train} = 14.11, \text{dev} = 4.83, \text{test} = 2.96)$.

4 Domain Learning

In the process of generating the NL2TS-675 dataset, the assignment of a domain to a group of time series is implicit and derived mainly from the textual description of an item. In our study, we propose a more natural approach and used machine learning for automating the allocation of time-series to a domain. To this end, we used time-series clustering with a heuristic algorithm that emphasizes both quality and execution performance. Rather than a static preprocess embedded in the dataset creation stage, we envisioned the grouping of temporal sequences a runtime problem, flexible to be conducted anytime a new set of time series is required for augmenting the data and subsequently each series allotted to a domain.

In Figure 1, we provide visualization of time-series clustering centroids. We employed k-means clustering along with Dynamic Time Warping (DTW) distance as the metric to measure similarity between two temporal sequences [11]. DTW between x and y is formulated as the following optimization problem:

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{(i,j)\in\pi} d(x_i, y_i)^2},$$

where π is the warping path for optimal alignment between the pair of variables. The computational complexity of DTW is quadratic in the lengths of the two input sequences m and n, $O(nm)$, and constructing a cost matrix $\in \mathbb{R}^{n\times m}$ has an $O(min(n, m))$ optimized space, when avoiding storage of the entire matrix. NL2TS-675 with its generated relatively short temporal sequences of less than 48 time steps, sustains a considerably reduced impact by the quadratic intensive math imposed by DTW, in contrast to longer sequences that span hundreds of time steps.

To validate our approach, we compared our clustering partition labels with the domain indices curated by humans, and achieved a score of 1.94 Mean Absolute Error (MAE) and 6.47 Mean Squared Error (MSE) on the test split of NL2TS-675. MAE and MSE metrics are inverted scoring interpreted as lower the better, and our results appear highly effective regardless of the underlying context.

In our evaluation, we also employed external clustering metrics, including 1) Adjusted Rand Index (ARI) [-1,1], 2) Normalized Mutual Information (NMI) [0,1], and 3) Purity (PuS) [0,1], with a respective score of 0.04, 0.13, and 0.49. In all, the higher the score the better. Commonly, a 0 score indicates poorly related and 1 identifies perfectly correlated clustering. Notably ARI lower bound can be negative and hence worse than random agreement. While the NMI and PuS measures are fairly compelling, ARI performance is non-commensurate most likely due to its sensitivity to noisy domain data.

5 Experiments

Task Definition. Given historical time series $X = x_{:L} \in \mathbb{R}^{L \times D}$ with L timesteps and D variables, time-series forecasting aims to infer the future states of H timesteps $\hat{X} = x_{:L+H} \in \mathbb{R}^{H \times D}$. We used Python notation for list slicing and let $X_{l,:}$ represent the series value at timestep l, and $X_{:,d}$ the entire time series for variable d.

Model Setup. In our study, we used Chronos-T5, a time series forecasting model that utilizes a small language model architecture to predict future data points from historical data. Chronos-T5 model uses 4,096 different tokens, compared to 32,128 of the original T5 models, resulting in much fewer parameters. We finetuned our chronos-t5-tiny checkpoint on the train and dev splits of the NL2TS-675 time-series dataset. In Table 2, we show runtime training and validation parameters for the finetuning process. The train set comprises 472 samples compared to 101 instances of the dev split (Table 1). We ran inference locally and entirely on the CPU with up to four workers, while not exceeding 8GB of system memory. At 6.16 iterations-per-second the running time on the entire NL2TS-675 test split of 102 time-series instances was at about 16 seconds.

Metrics. In our experiments, we applied the mean absolute error (MAE) and mean squared error (MSE)— common metrics used in forecasting to evaluate the accuracy of model predictions. The respective computations of the metrics follow: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$ and $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where n is the number of time steps in the prediction horizon, y_i and $\hat{y}_i$ are the i-th ground-truth and prediction values, respectively. The metrics use the same scale as the data being measured, thus taking the absolute or squared value ensures that all errors contribute positively to the final score.

Forecasting Quality. In table 3, we provide statistical domain distributions of MAE and MSE forecast scores evaluated on the NL2TS-675 test split. The scores presented are the average prediction accuracy of the time-series set for each domain, and each sequence traverses a different time length $L \in (12, 24, 48)$. To assist in our result interpretation, we provide a complementary Root Mean Squared Error (RMSE) measure column. The top single-digit MAE performance of 4.67 and 8.35 were gained by the weather and retail domains, respectively. This is consistent with their individually proportional RMSE rate, thus signifying more robustness to outliers. Conversely, due to the proliferation of edge devices and mobile sensing, the technology and healthcare domains were impacted the most on forecasting quality, as they demonstrate an extremely high standard deviation of the MSE scores and indicating a wide spread of time-series observations over a short-term range. Evidenced by our results, finance and healthcare significantly correlate on time-series forecasting. This mainly stems from an on-going integration of clinical and financial data and have both disciplines benefit from shared predictive analytic. Surprisingly the iot domain suffers less from outliers owing to its more predictable jagged pattern.

In Figure 2, we present forecast visualization examples for a time-series representative sample from each domain. The plots sustain qualitatively our quantifiable results as technology and healthcare share a similar signal shape. While the iot domain is fairly noisy it upholds only a slight deviation from the median trend.

Baseline Compare. In Table 4, we compare our forecast results with a baseline that uses a capable LLM (Llama 3.1) for inference. The baseline results were published by the authors of the NL2TS-675 dataset and only include MAE scores. Our model checkpoint for the chronos-t5-tiny has a significant performance lead for the weather and retail domains, about a double or triple increase, and a more moderate gain in iot. On the other hand, the baseline outperformed our model in healthcare by almost 2X, and we came fairly close to the baseline finance and technology measures. Overall, we showed an impressive two single-digit MAE scores and our global score has a slight edge over the baseline: 15.58 compared to 16.06.

Accuracy. In Table 6 (Appendix B), we provide measures of Mean Absolute Percentage Error (MAPE), a scale free metric that tells how far off predictions are from actual values. MAPE scores assess forecast accuracy with lower percentages representing better predictions. MAPE is computed as follows: $MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100$. MAPE scores below 10% is generally considered outstanding forecasting accuracy, 10%-20% is good, 20%-30% acceptable, 30%-50% poor and >50% is very poor. Our MAPE scores have the retail domain with 7% exceptionally ranked as excellent, finance and healthcare follow with a good rating, and iot and weather disciplines considered reasonably acceptable. The technology domain is trailing with 31% as borderline acceptable. Overall, our average MAPE score across all the six domains is 20%, which validates our scaled forecasting scores of good accuracy. However, the lack of published MAPE scores derived from the baseline inhibits a more complete performance comparison analysis.

6 Conclusion

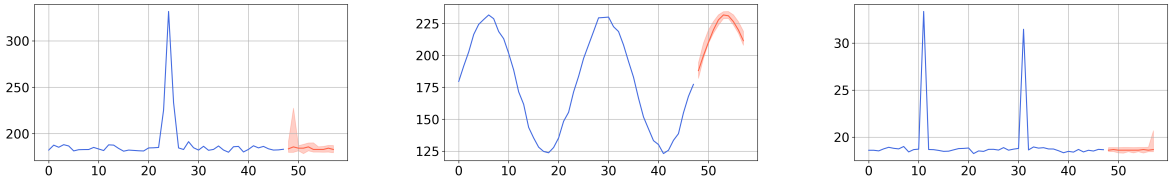
In this paper, we explored the use of a lightweight language model for the task of time-series forecasting on synthetically-generated temporal sequences. Overall, our global prediction score is consistent with a considerably more capable LLM. Across domains, our model outperformed the strong baseline on three of the six domains we experimented with,

Table 2: Runtime parameters on the train and dev splits of the NL2TS-675 dataset.

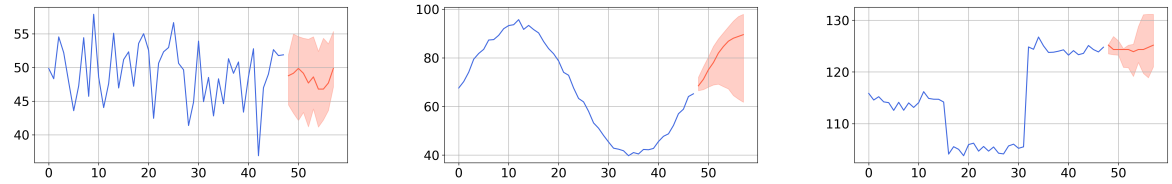
Split	Running Time (sec)	Samples/sec	Steps/sec	Loss	Epochs
Train	4,336.976	0.326	0.041	3.743	3
Dev	41.919	2.409	0.312	2.776	3

Table 3: Core statistical domain distributions of MAE, MSE, and RMSE inverted scoring— where a lower measure is better— on the NL2TS-675 test split. Top mean results are underlined.

		MAE			MSE			RMSE
Domain	Samples	Median	Mean	STD	Median	Mean	STD	Mean
Finance	32	16.31	22.22	17.15	341.88	962.26	1,266.58	31.02
Healthcare	11	12.70	21.74	19.34	228.08	1,084.27	1,683.26	32.92
Weather	17	4.19	<u>4.67</u>	3.54	40.79	<u>44.86</u>	52.93	<u>6.69</u>
IoT	11	8.63	14.37	15.22	94.42	485.16	805.74	22.02
Technology	13	14.62	22.15	24.39	280.46	1,231.22	2,156.97	35.08
Retail	18	7.81	8.35	5.86	97.67	134.42	133.58	11.59



(a) finance **(b) healthcare** **(c) weather**



(d) iot **(e) technology** **(f) retail**

Figure 2: Visualization of cross-domain forecasting samples for a history of 48 time steps (blue), a median forecast (red), and an 80% prediction interval (orange) for a horizon of 10 multi-steps on the test set of NL2TS-675.

Table 4: Domain mean MAE comparative inverted scoring on the NL2TS-675 test split. Lower score is better and top measures are underlined.

System	Finance	Healthcare	Weather	IoT	Technology	Retail	Overall
Ram [9]	<u>18.29</u>	<u>10.64</u>	15.04	16.21	<u>15.11</u>	16.91	16.06
Ours	22.22	21.74	<u>4.67</u>	<u>14.37</u>	22.15	<u>8.35</u>	<u>15.58</u>

but fell short over two domains where noise is prevalent. This is potentially due to the LLM sensitivity to noise when transforming the input time series from numbers to token representation. Our proposed framework for clustering temporal sequences at runtime rather than at dataset generation had proven highly effective for recurring dataset expansion.

7 Limitations

We acknowledge several limitations in our paper. We fine-tuned the Chronos-T5 model to better process numerical sequences by allowing LLMs to interpret time series data through natural language prompts. However, T5 is constrained to a fixed set of task-level prefixes added to the input sequence and instruct the model what task to perform. In its published version, NL2TS-675 supports short-term (12, 24, and 48) history for forecasting evaluations, and extending the dataset to a long-term history span (96 to 720) would benefit a standardized benchmark. Our work emphasizes low-cost and under-resourced for running inference entirely on the CPU. At the time of publication, Chronos-2 models [1] were less efficient on the CPU, although we anticipated forecasting quality to plausibly improve. Our framework adaptability to new data modalities remains an area for future exploration.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful suggestions and feedback.

References

[1] Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. 2025. Chronos-2: From Univariate to Universal Forecasting. arXiv:2510.15821 [cs.LG] https://arxiv.org/abs/2510.15821

[2] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. 2024. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=gerNCVqqtR Expert Certification.

[3] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2023. Large Language Models Are Zero-Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 19622–19635. https://proceedings.neurips.cc/paper_files/paper/2023/file/3eb7ca52e8207697361b2c0fb3926511-Paper-Conference.pdf

[4] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*.

[5] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation Models for Time Series Analysis: A Tutorial and Survey. In *SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 6555–6565. doi:10.1145/3637528.3671451

[6] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2022. Frozen Pretrained Transformers as Universal Computation Engines. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 7 (Jun. 2022), 7628–7636. doi:10.1609/aaai.v36i7.20729

[7] Jianyang Qin, Chaoyang Li, Jinhao Cui, Lingzhi Wang, Zhao Liu, and Qing Liao. 2025. Bridging Time and Linguistics: LLMs as Time Series Analyzer through Symbolization and Segmentation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=nOv6z9RHA5

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[9] Gokul Srinath Seetha Ram. 2025. Zero-to-Forecast: Natural Language to Time Series Prediction via Cross-Modal Ensembles. In *Recent Advances in Time Series Foundation Models Have We Reached the 'BERT Moment'?* https://openreview.net/forum?id=lERfNeDzul

[10] Cécile Rousseau, Tobia Boschi, Giandomenico Cornacchia, Dhaval Salwala, Alessandra Pascale, and Juan Bernabe Moreno. 2025. Forging Time Series with Language: A Large Language Model Approach to Synthetic Data Generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=A2pmvkqOgp

[11] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. 2020. Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research* 21, 118 (2020), 1–6. http://jmlr.org/papers/v21/20-091.html

[12] Qingsong Wen, Linxiao Yang, Tian Zhou, and Liang Sun. 2022. Robust Time Series Analysis and Applications: An Industrial Perspective. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 4836–4837. doi:10.1145/3534678.3542612

[13] Tian Zhou, Peisong Niu, xue wang, Liang Sun, and Rong Jin. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 43322–43355. https://proceedings.neurips.cc/paper_files/paper/2023/file/86c17de05579cde52025f9984e6e2ebb-Paper-Conference.pdf

A Time-Series Descriptor

Table 5: Modified JSON field format of an NL2TS-675 dataset example. The domain is ascribed post time-series clustering and the pattern attribute is explicit


```
{
  "uid": 101,
  "domain": "finance",
  "text": "Stock price shows gradual increase over time",
  "pattern": "trend",
  "series": [120.5, 125.2, 128.7, 132.1, 135.6, 138.9, 142.3, 145.1, 148.4,
  151.2, 154.8, 157.3],
  "freq": "hour",
  "length": 12,
} "split": "train"
```


B MAPE: scale-free metric

Table 6: Core statistical domain distributions of MAPE, on the NL2TS-675 test split. Top mean results are underlined.

		MAPE			
Domain	Samples	Median	Mean	STD	Range
Finance	32	0.16	0.16	0.08	0.29
Healthcare	11	0.13	0.17	0.14	0.52
Weather	17	0.23	0.26	0.24	0.87
IoT	11	0.21	0.23	0.20	0.55
Technology	13	0.21	0.31	0.35	1.13
Retail	18	0.05	<u>0.07</u>	0.05	0.21