Decoding Time Series with LLMs: A Multi-Agent Framework for Cross-Domain Annotation

Minhua Lin[1*], Zhengzhang Chen[2†], Yanchi Liu[2], Xujiang Zhao[2], Zongyu Wu[1], Junxiang Wang[2], Xiang Zhang[1], Suhang Wang[1], Haifeng Chen[2]
[1]The Pennsylvania State University [2]NEC Laboratories America
{mfl5681,zongyuwu,xzz89,szw494}@psu.edu
{zchen,yanchi,xuzhao,junwang,haifeng}@nec-labs.com

Abstract

Time series data is ubiquitous across various domains, including manufacturing, finance, and healthcare. High-quality annotations are essential for effectively understanding time series and facilitating downstream tasks. However, obtaining such annotations is challenging, particularly in mission-critical domains. In this paper, we propose TESSA, a multi-agent system designed to automatically generate both general and domain-specific annotations for time series data. TESSA introduces two agents: a general annotation agent and a domain-specific annotation agent. The general agent captures common patterns and knowledge across multiple source domains, leveraging both time-series-wise and text-wise features to generate general annotations. Meanwhile, the domain-specific agent utilizes limited annotations from the target domain to learn domain-specific terminology and generate targeted annotations. Extensive experiments on multiple synthetic and real-world datasets demonstrate that TESSA effectively generates high-quality annotations, outperforming existing methods.

1 Introduction

Time series data is prevalent in various fields such as climate (Chen et al., 2013; Li et al., 2025b), finance (Lee et al., 2024), and healthcare (Li et al., 2022). It captures critical temporal patterns essential for informed decision-making. However, general users frequently encounter difficulties in interpreting this data due to its inherent complexity, particularly in multivariate contexts where multiple variables interact over time. Furthermore, effective interpretation typically requires domain-specific knowledge to properly contextualize these patterns, thereby posing significant challenges for individuals without specialized expertise.

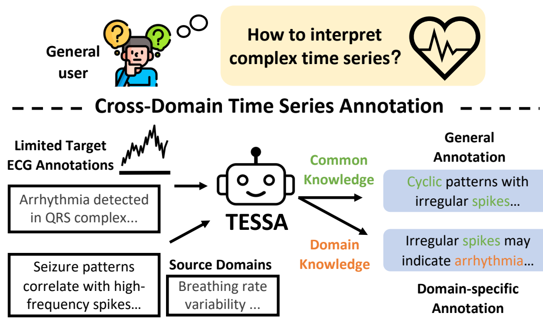


Figure 1: How to annotate time series automatically?

High-quality annotations are crucial for addressing these interpretive challenges. Annotations provide meaningful context or insights into time series data, highlighting important patterns, events, or anomalies. They facilitate accurate analysis, forecasting, and decision-making, enhancing the performance of downstream tasks such as anomaly detection, trend prediction, and automated reporting. For instance, in predictive maintenance, understanding sensor data trends is vital for preventing equipment failure, while in finance, interpreting stock price movements is crucial for informed investment strategies. Despite their importance, high-quality annotations are often scarce in real-world applications. This scarcity stems primarily from the reliance on domain experts for manual annotation, which is resource-intensive, costly, and prone to inconsistencies. Moreover, the need for precise and domain-specific terminology further complicates the annotation process, as different fields require highly specialized knowledge for accurate and contextually relevant interpretation.

To alleviate the above issues, one straightforward approach is to leverage external resources to generate annotations (Liu et al., 2024a). For example, Time-MMD (Liu et al., 2024a) uses web searches to retrieve information as annotations, aiming to find similar patterns and descriptions from the internet. Others (Jin et al., 2024; Liu et al., 2024b)

[*]Work done during an internship at NEC Labs America.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: EACL 2026, pages 6244–6281
March 24-29, 2026

directly adopt large language models (LLMs) for annotation, leveraging LLMs' great language understanding capability. Prototype-based methods, such as prototype networks (Ni et al., 2021), have also been employed to identify representative examples for annotation. However, *these methods often fall short of producing high-quality annotations*. Web search-based methods may retrieve irrelevant or inconsistent information. LLMs, while powerful, tend to generate generic annotations, capture only basic patterns, or even hallucinate, and fail to account for the complex nature of time series data. Prototype networks rely on large amounts of data to train the network and identify representative prototypes, but the scarcity of high-quality annotations limits the quality and representativeness of these prototypes, making it difficult to generalize effectively to new or unseen patterns.

To address these limitations, we propose to extract knowledge from existing annotations across multiple source domains and transfer this knowledge to target domains with limited annotations. Specifically, as shown in Fig. 1, we aim to develop a system that automatically interprets time series data across various fields using common or domain-specific language. Formally, given abundant annotations from multiple source domains and limited annotations from a target domain, our goal is to leverage both time-series-wise and text-wise knowledge to generate accurate and contextually appropriate annotations for the target domain. This raises two key technical challenges: **(i)** How to extract common knowledge from source domains? **(ii)** How to learn domain-specific jargon from limited target-domain annotations?

To tackle these challenges and overcome the limitations of existing methods, we propose **TESSA**, a multi-agent system designed for both general and domain-specific TimE SerieS Annotation. As illustrated in Fig. 2, TESSA introduces two agents: a general annotation agent and a domain-specific annotation agent. The general annotation agent focuses on capturing common patterns and knowledge across various domains to generate annotations understandable by general users. To learn common knowledge from multiple domains, the general agent employs a time series-wise feature extractor and a text-wise feature extractor to extract both time-series-wise and text-wise features from time series data and domain-specific annotations from multiple source domains. To ensure important features are included in the general annotations, two feature selection methods—LLM-based and reinforcement learning-based selection—are introduced to effectively and efficiently select both the top-k most important time-series-wise and text-wise features. The domain-specific agent leverages limited target-domain annotations to learn and generate annotations for specific domains using domain-specific terminologies (jargon). It incorporates a domain-specific term extractor to learn jargon from the limited target-domain annotations. Additionally, an annotation reviewer is proposed to maintain consistency between general annotations and domain-specific annotations.

Our contributions are: (i) **Problem**. We explore a novel problem in cross-domain multi-modal time series annotation, bridging the gap between general understanding and domain-specific interpretation; (ii) **Framework**. We propose a novel multi-agent system, TESSA, designed for both general and domain-specific time series annotation by leveraging both time-series-wise and text-wise knowledge from multiple domains; (iii) **Datasets**. We collect a real-world dataset from finance domain to leverage cross-domain knowledge, along with a synthetic dataset to evaluate TESSA. (iv) **Experiments**. Extensive experiments on multiple synthetic and real-world datasets demonstrate the effectiveness of TESSA in producing high-quality annotations.

2 Related Work

Time Series Annotation. Time series annotation aims to assign labels or descriptions to specific segments, events, or patterns within a time series dataset to highlight significant features for further analysis. Traditionally, this process has relied on manual annotation (Reining et al., 2020), which is often time-consuming, labor-intensive, and requires substantial domain expertise. To reduce the effort needed for creating large-scale, high-quality annotated datasets, several studies have proposed semi-automatic annotation approaches (Cruz-Sandoval et al., 2019; Nino et al., 2016) that require minimal manual input or post-annotation revisions. Despite these advancements, fully automated time series annotation remains underexplored due to the challenges of capturing semantic and contextual information from the data (Yordanova and Krüger, 2018).

LLMs for Time Series Analysis. Recent advancements in LLMs have showcased their strong capabilities in sequential modeling and pattern recog-

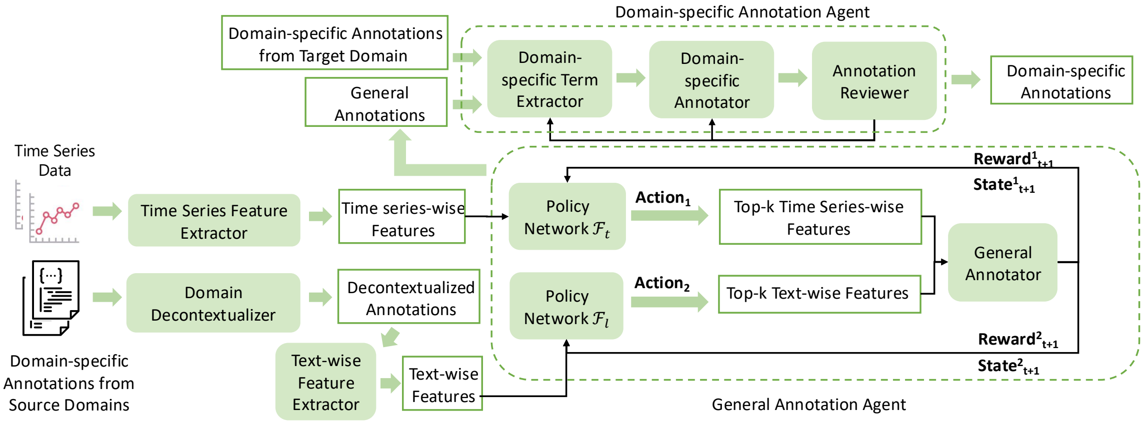


Figure 2: Overall framework of TESSA. It consists of two main agents: a general annotation agent, which generates domain-independent annotations by selecting salient time-series and textual features, and a domain-specific annotation agent, which refines these annotations by incorporating domain-specific terminology.

nition, opening up promising new directions for time series analysis. Several studies (Shen et al., 2025; Xue and Salim, 2023; Yu et al., 2023; Gruver et al., 2024; Jin et al., 2024; Li et al., 2024) have explored this potential. For instance, PromptCast (Xue and Salim, 2023) is a pioneering work that applies LLMs to general time series forecasting using a sentence-to-sentence approach. Time-LLM (Jin et al., 2024) reprograms time series into textual prototypes for LLaMA-7B, enhanced by natural language prompts incorporating expert knowledge. More recently, retrieval-augmented designs have been proposed for *zero-shot forecasting*. TS-RAG (Ning et al., 2025) retrieves semantically relevant time-series segments from a knowledge base and fuses them with a forecasting backbone to improve robustness and interpretability. In contrast, BRIDGE (Li et al., 2025a) focuses on text-controlled time-series generation, using natural-language descriptions and an LLM-based multi-agent pipeline to synthesize paired text–time-series data. Unlike these forecasting- or generation-oriented approaches, our work targets cross-domain time-series annotation, aiming to produce general and domain-specific semantic annotations that support interpretability and downstream tasks. Additional related work is discussed in Appendix A.1.

Cross-modality Knowledge Transfer Learning through Pre-trained Models. There has been growing interest in leveraging pre-trained models for cross-modality knowledge transfer, particularly between the language, vision, and time series domains (Bao et al., 2022; Lu et al., 2022; Wang et al., 2024; Yang et al., 2021; Zhou et al., 2023). Recently, Zhou et al. (2023) have applied pre-trained language and image models to time series analysis tasks. To the best of our knowledge, no previous work has specifically explored cross-modality knowledge transfer for time series annotation. Our work fills this gap by exploring how cross-modality transfer learning can enable automatic time series annotation. More details on the related work are provided in the Appendix A.3

3 Methodology

In this section, we define the problem and present the details of our proposed TESSA framework, which aims to generate both general and domain-specific annotations for time series data.

Cross-Domain Time Series Annotation Problem. Given several source domains $\{\mathcal{D}_{s_1}, \mathcal{D}_{s_2}, \dots\}$ and a target domain $\mathcal{D}_t$, let $\{e^1_{s_i}, e^2_{s_i}, \dots\}$ denote the domain-specific annotations from the source domain $\mathcal{D}_{s_1}$, and $\{e^1_t, e^2_t, \dots\}$ represent the limited domain-specific annotations from the target domain $\mathcal{D}_t$. Suppose $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_L)$ is a time series in $\mathcal{D}_t$, where L is the number of past timestamps and $\mathbf{x}_i = (x_{1i}, \cdots, x_{Ci})^T \in \mathbb{R}^C$ represents the data from C different channels at timestamp i. The objective of cross-domain time series annotation is to generate the general annotation e_g and the domain-specific annotation e_s for $\mathbf{X}$ based on the annotations from both the source and target domains. More notations are provided in Appendix B.

Overview of TESSA. As illustrated in Fig. 2, the proposed TESSA comprises two key components: a general annotation agent and a domain-specific annotation agent. The general annotation agent is responsible for generating domain-independent annotations and consists of several modules: a time series feature extraction module to capture time-series-specific features, a domain decontextualization module to convert domain-specific text into common language, a text feature extraction module to retrieve textual features from the decontextualized text, two policy networks for selecting the top-k most salient time-series and textual features, and a general annotator to produce general annotations based on the selected features. The domain-specific annotation agent refines the general annotations to generate domain-specific annotations. It includes a domain-specific term extractor to identify key terminology from a limited set of target-domain annotations and a domain-specific annotator to adjust the general annotations accordingly. An annotation reviewer further enhances the quality of the domain-specific annotations. Next, we introduce details of each component.

3.1 Multi-modal Feature Extraction

To address the challenge of extracting common knowledge from source domains, we introduce two feature extraction modules: a time-series feature extractor and a text-wise feature extractor, which extract features from time series data and source-domain annotations. We also propose a domain decontextualizer to enhance the extraction of common knowledge from multi-source annotations.

Time Series Feature Extraction. We extract features from time series data through a structured process $\mathcal{M}_r$. Formally, for each channel $c \in C$, the set of time-series features $\mathbf{F}_t$ is denoted as:

$$\mathbf{F}_t = \{f_t^1, \cdots, f_t^{n_t}\} = \mathcal{M}_r(\mathbf{X}), \tag{1}$$

where $\mathcal{M}_r$ denotes the feature extraction framework applied to $\mathbf{X}$, f_t^i is the i-th extracted feature of $\mathbf{X}$, and n_t is the number of extracted features. For multivariate time series data, inter-variable features (*e.g.*, Pearson correlation) are also included. Details of $\mathcal{M}_r$ are provided in Appendix C.

Domain Decontextualization. In addition to time-series-wise features, textual annotations from source domains often contain valuable information (such as support or resilience in finance time series annotations) for interpreting time series data. A straightforward method to extract this knowledge is to use LLMs on domain-specific annotations, leveraging their real-world knowledge. However, in practice, many domains lack sufficient high-quality annotations, and domain-specific terminology can further hinder effective extraction.

To address these challenges and facilitate knowledge transfer from source to target domains, we introduce a domain decontextualization LLM to convert domain-specific annotations into general annotations by removing domain-specific terminology. This makes it easier to extract common knowledge across domains. Specifically, given a domain-specific annotation e_s^i in domain d_i, the decontextualized annotation e_d^i is obtained as:

$$e_d^i = \mathcal{M}_d(p_{de}(e_s^i, d_i)), \tag{2}$$

where $\mathcal{M}_d$ is the domain decontextualization LLM. Details of the prompt template p_{de} and examples can be found in Appendix C.2.

Text Feature Extraction. After decontextualization, we use an LLM $\mathcal{M}_l$ to extract textual features from multiple source domains. Formally, given a set of decontextualized annotations $\{e_d^i\}_{i=1}^{n_d}$ and the text feature extractor $\mathcal{M}_l$, the extracted textual features are denoted as:

$$\mathbf{F}_l = \{f_l^1, \cdots, f_l^n\} = \mathcal{M}_l(p_l(\{e_d^i\}_{i=1}^{n_d})), \tag{3}$$

where p_l is the prompt for text feature extraction to guide $\mathcal{M}_l$ to output the text-wise features explicitly or implicitly mentioned in the decontextualized annotations. The complete prompt template is provided in Appendix C.3.

3.2 Adaptive Feature Selection

With a diverse set of features extracted from time series and text data, it becomes essential to focus on the most relevant ones to ensure the generated annotations remain concise and interpretable. Moreover, repeatedly querying LLMs with both the old and new data[1] each time wastes computational resources and incurs additional costs, especially when using non-open-source models.

To address these issues, we propose a hybrid strategy for adaptive feature selection that combines *Offline LLM-based Feature Selection* with *Incremental Reinforcement Learning-based Feature Selection*. The incremental method builds on the offline approach, minimizing the need to re-query LLMs with both old and new data as it arrives.

Offline LLM-based Feature Selection. Leveraging LLMs' reasoning abilities, we introduce a feature selection method using LLM-generated feature importance scores to identify the top-k most

[1]To avoid redundancy, unless specified otherwise, 'data' in this paper refers to time series and their corresponding textual annotations from various domains.

important time-series-wise and text-wise features. Features mentioned more frequently—either explicitly or implicitly—in annotations are assigned higher importance scores.

Specifically, given an LLM as the feature selector $\mathcal{M}_{sel}$, we prompt $\mathcal{M}_{sel}$ with domain-decontextualized annotations $\{e_d^i\}_{i=1}^{n_d}$ and the extracted features $\{f_t^i\}_{i=1}^{n_t}$ and $\{f_l^i\}_{i=1}^{n_l}$ to generate numerical feature importance scores: $\mathbf{s}_t = [s_1, \cdots, s_{n_t}]$ for time-series-wise features and $\mathbf{s}_l = [s_1, \cdots, s_{n_l}]$ for text-wise features.

$$\begin{aligned} s_j &= \mathcal{M}_{sel}(p_{score}(f_t^j, \{e_d^i\}_{i=1}^{n_d})), \ \ \forall j \in \{1, \cdots, n_t\}, \\ s_k &= \mathcal{M}_{sel}(p_{score}(f_l^k, \{e_d^i\}_{i=1}^{n_d})), \ \ \forall k \in \{1, \cdots, n_l\}, \end{aligned} \tag{4}$$

Here, p_{score} is the prompt used to score feature importance. Higher scores, s_j and $s_k \in \mathbb{R}^+$, indicate that the features f_t^j and f_l^k appear more frequently, either explicitly or implicitly, in the domain-decontextualized annotations $\{e_d^i\}_{i=1}^{n_d}$. The templates for p_{score} are shown in Fig. 14 and Fig. 15, respectively. To ensure that explicitly mentioned features receive higher importance scores, we instruct $\mathcal{M}_{sel}$ to assign greater weight to features that are explicitly referenced in the annotations. More details are provided in Appendix D.1.

Incremental Reinforcement Learning-based Feature Selection. When new data[1] arrives, the offline LLM-based approach requires re-querying both old and new data, which becomes burdensome due to LLMs' limited context window. As annotations increase, re-querying all data becomes impractical and costly, leading to higher resource consumption and reduced cost-effectiveness.

To address the limitations of the offline approach, we propose an *Incremental Reinforcement Learning-based Feature Selection* method that is more cost-effective for dynamic environments with evolving data. Specifically, we introduce a multi-agent reinforcement learning (MARL) framework to train two policy networks, $\mathcal{F}_t$ and $\mathcal{F}_l$, to select the top-k most important time-series-wise and text-wise features, respectively. These policy networks store knowledge from existing annotations and are incrementally updated as new data arrives. This reduces the need to re-query the LLM with all the data, requiring only the new data during updates. As shown in Fig. 2, each policy network is initialized with the first three layers of a small LLM, such as GPT-2 (Radford et al., 2019), which remain frozen during training. A trainable multi-head attention layer and a language model (LM) head from GPT-2 follow these layers, using the smallest version of GPT-2 with 124M parameters.

During training, only the multi-head attention layer is updated. For time-series-wise features, given the candidate features $\{f_t^i\}_{i=1}^{n_t}$ and their corresponding feature name tokens $\mathbf{Y} = \{y_1^i, \cdots, y_{n_t}^i\}$, the policy network $\mathcal{F}_t$ computes action-values (Q-values) $\mathbf{q}_z = [q_{z,f_t^1}, \cdots, q_{z,f_t^{n_t}}]$ based on the mean logits of the feature names:

$$\mathbf{q}_s = \mathcal{F}_t(\{y_i\}_{i=1}^{n_t}), \tag{5}$$

A softmax function generates a probability distribution over the features, and the top-k features are selected based on the highest probabilities.

At each timestep, the selected top-k features are passed to the LLM $\mathcal{M}_{sel}$ to obtain their importance scores $s_i, \forall i \in \{1, \cdots, k\}$. The agent receives a reward r_t defined as:

$$r_t = \begin{cases} \sum_{i=1}^{k} s_i, & s_i \geq \tau \\ -0.5, & \text{otherwise,} \end{cases} \tag{6}$$

where τ is a threshold to discourage selecting unimportant features. The text-wise feature policy network $\mathcal{F}_l$ undergoes a similar training process.

After training, the policy networks are incrementally updated with only new data, eliminating the need to re-query the LLM with both old and new data. This approach improves the scalability and efficiency of feature selection while reducing computational costs, effectively overcoming the offline approach's limitations. By incrementally updating the policy networks, we ensure that feature selection remains scalable and cost-effective in dynamic environments with evolving data. More discussion of the necessity of the RL component is provided in Appendix D.2.

3.3 General Annotation Generation

After selecting the top-k most important features from both time-series and text, a general annotator is introduced to generate general annotations by analyzing these selected features. An LLM, serving as the general annotator, interprets the given time series data based on the selected features. Formally, given time series data $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{L}$ and the selected time-series-wise and text-wise features $\{f_t^i\}_{i=1}^{k_t}$ and $\{f_l^i\}_{i=1}^{k_l}$, the generation of a general annotation e_g is represented as:

$$e_g = \mathcal{M}_{gen}(p_{gen}(\{\mathbf{x}_i\}_{i=1}^{L}, \{f_t^i\}_{i=1}^{k_t}, \{f_l^i\}_{i=1}^{k_l})), \tag{7}$$

where p_{gen} is the prompt for generating general annotations. By emphasizing the signal from the selected common knowledge, the general annotations capture richer patterns that may be overlooked when directly applying LLMs. An example of the prompt template is shown in Fig. 16.

3.4 Domain-specific Annotation Generation

Generating domain-specific annotations for time series is crucial as different domains rely on specialized jargon and context-specific terminology to accurately interpret and understand data. Time series data from financial markets, healthcare systems, or industrial processes can exhibit patterns, trends, and anomalies that are unique to each domain. General annotations may overlook critical nuances, whereas domain-specific annotations capture contextual relevance, improving the precision and reliability of downstream analysis or model predictions. By tailoring annotations to a domain's specific lexicon, we can detect meaningful patterns more accurately and make informed decisions.

Domain-specific Term Extractor. To address the challenge of learning domain-specific terminology, we introduce a domain-specific term extractor. Given limited domain-specific annotations $\{e_t^i\}_{i=1}^{n_{e_t}}$ from the target domain, an LLM $\mathcal{M}_{ext}$ is employed to extract domain-specific terms. We prompt $\mathcal{M}_{ext}$ with the annotations $\{e_t^i\}_{i=1}^{n_{e_t}}$ to extract a set of domain-specific terms $\{\mathcal{G}^i\}_{i=1}^{n_g}$:

$$\{\mathcal{G}^i\}_{i=1}^{n_g} = \mathcal{M}_{ext}(p_{ext}(\{e_t^i\}_{i=1}^{n_{e_t}})), \tag{8}$$

where n_g is the number of extracted terms, and p_{ext} is the prompt for domain-specific term extraction. Fig. 17 provides the template for p_{ext}.

Domain-specific Annotator. To ensure alignment between domain-specific and general annotations, an LLM $\mathcal{M}_{spe}$, acting as a domain-specific annotator, applies the extracted terms $\{\mathcal{G}^i\}_{i=1}^{n_g}$ to general annotations e_g, converting them into target-domain annotations e_t. Formally, this is represented as:

$$e_t = \mathcal{M}_{spe}(p_{spe}(e_g, \{\mathcal{G}^i\}_{i=1}^{n_g})), \tag{9}$$

where p_{spe} is the prompt for generating domain-specific annotations, shown in Fig. 18.

Annotation Reviewer. To improve the quality of domain-specific annotations and ensure better alignment with general annotations, we introduce an annotation reviewer. This LLM, $\mathcal{M}_{rev}$, reviews the generated annotations and extracted terms, providing feedback e_f to the extractor and annotator:

$$e_f = \mathcal{M}_{rev}(p_{rev}(e_g, e_t, \{\mathcal{G}^i\}_{i=1}^{n_g})), \tag{10}$$

where p_{rev} is the prompt for reviewing annotations. An example is shown in Fig. 19. This feedback loop ensures more precise term extraction and better alignment between general and domain-specific annotations. Based on the feedback, the extractor $\mathcal{M}_{ext}$ refines the extraction process, and the annotator $\mathcal{M}_{spe}$ enhances its annotations accordingly.

4 Experiments

This section presents the experimental results. We first evaluate the TESSA's annotations in downstream tasks and on a synthetic dataset, then examine domain-specific annotations, and finally assess the contribution of key TESSA components.

4.1 Experimental Setup

Dataset. To evaluate the effectiveness of TESSA, five real-world datasets from distinct domains are considered: Stock, Health, Energy, Environment, Social Good, Climate and Economy. Specifically, the stock dataset includes 1,935 US stocks with the recent 6-year data, collected by ourselves. The other four datasets come from the public benchmark Time-MMD (Liu et al., 2024a). In this paper, the Stock and Health datasets serve as the source domains, while the rest five datasets are treated as the target domains. Additionally, we generate a synthetic dataset containing both time series and ground-truth annotations to directly assess the quality of general annotations. More details on these datasets can be found in Appendix F.1.

LLMs. Our experiments utilize one closed-source model, GPT-4o (Achiam et al., 2023) and two open-source models, LLaMA3.1-8B (Dubey et al., 2024) and Qwen2-7B (Yang et al., 2024).

4.2 Evaluating General Annotations in Downstream Tasks

To evaluate the quality of the general annotations, we apply the generated annotations to the multi-modal downstream tasks (*i.e.*, time series forecasting and imputation) by following the experimental setup in Time-MMD (Liu et al., 2024a). As in Fig. 4, time series data and textual annotations are processed independently by unimodel TSF models and LLMs with projection layers. The model-specific outputs are then fused through a linear weighting mechanism to generate final predictions. Incremental RL-based selection in Section 3.2 is used in TESSA to select the top-k most important features for generating annotations. The implementation details are provided in Appendix F.3 and G.1.

Table 1: Forecasting results with GPT-4o as the LLM backbone. NT, TM, and DL refer to No-Text, Time-MMD, and DirectLLM, respectively. MSE is shown in the top half and MAE in the bottom half.

Domain	NT	TM	DL	TESSA
Environment	1.2542	0.8483	0.7714	**0.4629**
Energy	2.0117	0.2172	0.0575	**0.0482**
Social Good	2.1457	1.6072	0.4639	**0.1935**
Environment	0.7387	0.6865	0.6604	**0.4424**
Energy	1.1663	0.2139	0.0055	**0.0040**
Social Good	1.1205	0.9731	0.3801	**0.0825**

Baselines. TESSA is, to the best of our knowledge, the first work on cross-domain multi-modal time series annotation. We compare it with several representative single-domain methods: No-Text, Time-MMD (Liu et al., 2024a), and DirectLLM (which directly uses LLM-generated annotations). Details of these methods are provided in Appendix F.2.

Evaluation Metrics. For time series forecasting task, we use MSE (Mean Squared Error) and MAE (Mean Absolute Error) as evaluation metrics, where lower values for them mean better annotations.

Experimental Results. Table 1 presents the comparison results for the time series forecasting task, where Informer (Zhou et al., 2021) is the forecasting model and GPT-4o (Achiam et al., 2023) serves as the LLM backbone. Additional forecasting results using different LLM backbones are available in Appendix G.2. The following observations can be made: (1) No-Text shows the worst performance across all datasets, validating the need for annotations to improve performance in downstream tasks. This suggests that better downstream task performance indicates higher-quality annotations. (2) TESSA achieves the best performance among all compared methods, demonstrating its effectiveness in generating high-quality general annotations. Additional results of time series imputation tasks, can be found in Appendix G.3.

4.3 Evaluating General Annotations in Synthetic Datasets

We construct a synthetic dataset with time series data and ground-truth annotations to validate TESSA's performance. Implementation details are provided in Appendix H.1.

Evaluation Metrics. We apply the LLM-as-a-judge approach (Bubeck et al., 2023; Dubois et al., 2024), evaluating two metrics: Clarity and Comprehensiveness. Two distinct LLMs score the generated annotations on a scale of 1 to 5 for each metric, with an overall score calculated as the mean of the two metrics. Further details on the metrics and the LLM-judge prompts can be found in Appendix H.2.

Table 2: General annotation results on the synthetic dataset with GPT-4o as the LLM backbone.

Metric	Method	Mean	P(T>D) (%)
Clarity	TESSA	**3.90**	**69.76**
	DirectLLM	3.79	
Compre.	TESSA	**4.44**	**87.10**
	DirectLLM	1.55	
Overall	TESSA	**4.14**	**82.71**
	DirectLLM	2.84	

Experimental Results. We compare TESSA with DirectLLM in Table 2. The "Mean" denotes the average score of generated annotations for each method, and **P(T>D)** is the percentage of TESSA's annotations that receive higher scores than DirectLLM's. The results show that TESSA outperforms DirectLLM on both metrics, with average scores of 3.90 in Clarity and 4.44 in Comprehensiveness, compared to DirectLLM's 3.79 and 1.55. Additionally, 82.71% of TESSA's annotations receive higher scores, indicating that TESSA produces more essential and easily understandable features, further demonstrating its effectiveness.

4.4 Domain-specific Annotation Evaluation

In this subsection, we evaluate the quality of domain specific annotations. Similar to Section 4.3, we adopt a LLM-as-a-Judger strategy to evaluate the performance of domain-specific annotation agent from three perspectives: Clarity, Comprehensiveness, and Domain-relevance. The overall score is the average of these three metrics. Further details on these metrics are provided in Appendix I.1.

Experimental Results. We present the comparison results of TESSA and DirectLLM on the Environment dataset in Table 3, with GPT-4o as the LLM backbone. The key observations are: (1) TESSA significantly outperforms DirectLLM across all metrics, achieving an overall score of 4.64 compared to DirectLLM's 3.41. Notably, 98.51% of TESSA's annotations receive higher scores, demonstrating its effectiveness in generating high-quality domain-specific annotations. (2)

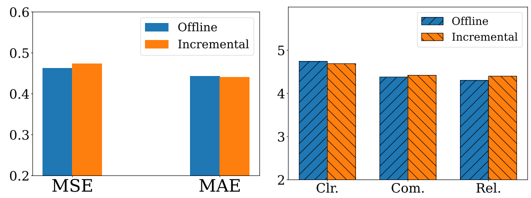
Table 3: Domain-specific annotation results on the Environment dataset using GPT-4o as the LLM. Dom. Rel. is the domain-relevance metric used in Section 4.4.

Metric	Method	Mean	P(T>D) (%)
Clarity	TESSA	**4.74**	**99.81**
	DirectLLM	3.32	
Compre.	TESSA	**4.38**	**97.04**
	DirectLLM	3.01	
Dom. Rel.	TESSA	**4.30**	**94.72**
	DirectLLM	3.57	
Overall	TESSA	**4.64**	**98.51**
	DirectLLM	3.41	

TESSA scores 4.74 in Clarity and 4.38 in Comprehensiveness, while DirectLLM scores 3.32 and 3.01, respectively. This shows that TESSA 's annotations are clearer, more concise, and cover more important features. (3) TESSA also excels in domain relevance, with 94.72% of its annotations scoring higher, achieving an average of 4.30, significantly outperforming DirectLLM's 3.41. This indicates that TESSA produces highly accurate annotations that effectively use domain-specific terminology and maintain strong contextual relevance. More results on other datasets are in Appendix I.3. We also conduct human-in-the-loop validate to further demonstrate that the annotations generated by TESSA can assist humans in analyzing time-series data, where more details are in Appendix J.

4.5 In-depth Dissection of TESSA

Adaptive Feature Selections. We compare our two feature selection methods: offline LLM-based selection and incremental RL-based selection. To assess their effectiveness in selecting the top-k most important features, we evaluate the quality of the generated general and domain-specific annotations, following the procedures in Sections 4.2 and 4.4. Environment is set as the target domain, with results shown in Fig. 3. The results indicate that TESSA performs comparably in both general and domain-specific annotation generation using either selection method. Specifically, as shown in Fig. 3(a), both approaches achieve MSE and MAE around 0.46 and 0.44 for general annotations. Similarly, in Fig. 3(b), both methods score consistently high across all domain-specific metrics, demonstrating their effectiveness in selecting important features. However, incremental RL-based selection



(a) General (b) Specific

Figure 3: Comparison of offline vs. incremental feature selection. GPT-4o is the LLM backbone, with Environment as the target domain. (a) General annotation results; (b) Domain-specific annotation results.

proves more cost-effective by reducing redundant re-querying of previously used data.

Ablation Studies. We perform ablation studies to assess the importance of domain decontextualization and adaptive feature selection in TESSA. For domain decontextualization, we introduce a variant, TESSA/D, which bypasses the domain decontextualization LLM and directly extracts text-wise features from domain-specific annotations. Table 19 shows that TESSA/D captures irrelevant features, such as *higher prices over time* and *fun*, which are unrelated to time series analysis. This confirms that domain-specific terminology can hinder the accurate extraction of time-series-relevant features.

To prove the importance of adaptive feature selection in TESSA, we remove the adaptive feature selection module to create a variant, TESSA/F. We apply an LLM-as-a-judger to compare the quality of the generated annotations between TESSA and its variants. The evaluation metrics are introduced in Appendix L.1. The comparison results on the Social Good dataset are in Table 5, with qualitative examples in Appendix L.2. We observe: TESSA consistently outperforms TESSA/F. Specifically, TESSA achieves a clarity score of 4.41, compared to 3.66 for TESSA/F. This demonstrates the necessity of adaptive feature selection. Furthermore, according to Table 21 in Appendix L.2, the annotations generated by TESSA/F tend to include many features without proper analysis. This shows that involving too many features can hinder the clarity of the annotations, further emphasizing the importance of adaptive feature selection in improving annotation quality. Additional ablation studies examining the contributions of other components of TESSA are in Appendix L.3. And discussions on data contamination are in Appendix L.4.

Comparison with Multi-agent Systems. To better contextualize TESSA's contributions against general-purpose multi-agent frameworks, such as AutoGen (Wu et al., 2024), MetaGPT (Hong et al., 2024), CAMEL (Li et al., 2023), we argue that while general-purpose systems like AutoGen, MetaGPT, and CAMEL have shown promise in task planning and code generation, *they are not tailored for time series annotation*. To evaluate their suitability, we conducted a new experiment by adapting AutoGen for our task.

To ensure a fair comparison, we configured AutoGen with the same backbone and prompts as TESSA. Following the setup in Section 4.4, we evaluated all models on the SocialGood dataset using our LLM-as-a-Judge framework. Results in Table 10 show that while AutoGen improves upon DirectLLM, it underperforms TESSA. This gap arises because AutoGen's general sequential coordination lacks TESSA's specialized architectural features: LLM-guided feature selection (Sec. 3.1) for robust cross-domain patterns, a Domain-Specific Term Extractor (Sec. 3.3) to leverage jargon from sparse annotations, and a bi-directional Annotation Reviewer loop (Sec. 3.4) for iterative refinement. Ultimately, this highlights that specialized agent roles and domain adaptation mechanisms are essential for high-quality time series annotation. More details of the comparison are in Appendix F.5.

Strengthened Baseline Evaluations. Beyond multi-agent systems, we substantially expand our baseline comparisons to ensure a rigorous evaluation. We assess annotation utility using strong forecasting backbones, including PatchTST and Reformer (Appendix F.7), together with a joint fine-tuning baseline on combined source and target data (Appendix F.4). To examine whether TESSA remains effective when the LLM backbone has no exposure to time-series–specific data, we conduct an ablation study using a fully open model, OLMo-7B (Groeneveld et al., 2024) (Appendix L.4). We further evaluate generalizability on both the SocialGood benchmark and the traditional ETT-small_h1 (Zhou et al., 2021) dataset (Appendix F.6). Collectively, these results demonstrate the effectiveness of TESSA for high-quality time-series annotation.

4.6 Case Study of TESSA

We conduct a case study to further validate the effectiveness of TESSA. A representative time series from the Social Good domain (Fig. 6(b)) is selected, and both TESSA and DirectLLM are applied to generate general and domain-specific annotations, summarized in Table 28. To assess the quality of the annotations, we use an LLM-as-a-judger to evaluate the domain-specific annotations from both methods, with results shown in Table 20. Our findings indicate that: (1) TESSA's general annotations capture more meaningful patterns, aiding user understanding and downstream tasks, whereas DirectLLM only highlights basic trends; and (2) TESSA's domain-specific annotations consistently outperform DirectLLM across all metrics, offering clearer, more comprehensive, and contextually relevant insights. More case studies of multivariate time series data are provided in Appendix M.

Table 4: Comparison between TESSA and AutoGen on domain-specific annotation. The target dataset is SocialGood, with Qwen2-7B as the LLM backbone. Stock and Health serve as the source datasets.

Metric	DirectLLM	AutoGen	TESSA
Clarity	3.28	3.89	**4.68**
Compre.	3.26	3.93	**4.49**
Dom. Rel.	3.33	4.01	**4.45**
Overall	3.29	3.94	**4.48**

Table 5: Ablation studies in the SocialGood dataset. GPT-4o is the LLM backbone.

Metric	Method	Mean	P(T>D) (%)
Clarity	TESSA	**4.41**	**83.3**
	TESSA/F	3.66	

5 Conclusion

In this work, we introduce TESSA, a multi-agent system for automatic general and domain-specific time series annotation. TESSA incorporates two agents, a general annotation agent and a domain-specific annotation agent, to extract and leverage both time-series-wise and text-wise knowledge from multiple domains for annotations. TESSA overcomes the limitations of directly applying LLMs, which often capture only basic patterns and may hallucinate, by effectively identifying and emphasizing significant patterns in time series data. Our experiments on synthetic and real-world datasets from diverse domains demonstrate the effectiveness of TESSA in generating high-quality general and domain-specific annotations.

6 Limitations

Potential limitations of this work include the need for limited target-domain annotations to learn domain-specific jargon for generating domain-specific annotations. Additionally, our approach relies on annotations from source domains to transfer knowledge to target domains. If the chosen source domain annotations are all of low quality or lack sufficient common knowledge, it may affect the overall performance of TESSA.

7 Ethics Statement

We adhere to the ACM Code of Ethics in our research. All datasets and models used in this study are either publicly accessible or synthetically generated. Specifically, we created a synthetic dataset comprising time series data with generated general annotations to facilitate our experiments while avoiding the use of any personal or sensitive real-world data. We acknowledge the potential risks and harms associated with LLMs, such as generating harmful, offensive, or biased content. Moreover, LLMs are often prone to generating incorrect information, sometimes referred to as hallucinations. We recognize that the models studied in this paper are not exceptions to these limitations. Previous research has shown that the LLMs used in this study suffer from bias, hallucinations, and other issues. We emphasize the importance of responsible and ethical use of LLMs and the need for further research to mitigate these challenges before deploying them in real-world applications. The models used in this work are licensed under the terms of OpenAI, LLaMA, and Qwen.

References

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

Luis B Almeida. 1994. The fractional fourier transform and time-frequency representations. *IEEE Transactions on signal processing*, 42(11):3084–3091.

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. 2022. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, pages 32897–32912.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. *Transactions of the Association for Computational Linguistics*, 10:414–433.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2024. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. In *ICLR*.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *arXiv preprint arXiv:2308.08469*.

Zhengzhang Chen, William Hendrix, Hang Guan, Isaac K Tetteh, Alok Choudhary, Fredrick Semazzi, and Nagiza F Samatova. 2013. Discovery of extreme events-related communities in contrasting groups of physical system networks. *Data Mining and Knowledge Discovery*, 27(2):225–258.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. 1990. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73.

Dagoberto Cruz-Sandoval, Jessica Beltran-Marquez, Matias Garcia-Constantino, Luis A. Gonzalez-Jasso, Jesus Favela, Irvin Hussein Lopez-Nava, Ian Cleland, Andrew Ennis, Netzahualcoyotl Hernandez-Cruz, Joseph Rafferty, Jonathan Synnott, and Chris Nugent. 2019. Semi-automated data labeling for activity recognition in pervasive healthcare. *Sensors*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. International Conference on Learning Representations, ICLR.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Yushan Jiang, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. Empowering time series analysis with large language models: A survey. *arXiv preprint arXiv:2402.03182*.

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 *ICLR*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. 2024. A survey of large language models in finance (finllms). *arXiv preprint arXiv:2402.02315*.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Hao Li, Yu-Hao Huang, Chang Xu, Viktor Schlegel, Renhe Jiang, Riza Batista-Navarro, Goran Nenadic, and Jiang Bian. 2025a. BRIDGE: Bootstrapping text to control time-series generation via multi-agent iterative optimization and diffusion modeling. In *Forty-second International Conference on Machine Learning*.

Jun Li, Che Liu, Sibo Cheng, Rossella Arcucci, and Shenda Hong. 2024. Frozen language model helps ecg zero-shot learning. In *Medical Imaging with Deep Learning*, pages 402–415. PMLR.

Quan Li, Wenchao Yu, Suhang Wang, Minhua Lin, Lingwei Chen, Wei Cheng, and Haifeng Chen. 2025b. xtime: Extreme event prediction with hierarchical knowledge distillation and expert fusion. In *ICDM 2025*.

Yuening Li, Zhengzhang Chen, Daochen Zha, Mengnan Du, Jingchao Ni, Denghui Zhang, Haifeng Chen, and Xia Hu. 2022. Towards learning disentangled representations for time series. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3270–3278.

Minhua Lin, Hui Liu, Xianfeng Tang, Jingying Zeng, Zhenwei Dai, Chen Luo, Zheng Li, Xiang Zhang, Qi He, and Suhang Wang. 2025a. How far are llms from real search? a comprehensive study on efficiency, completeness, and inherent capabilities. *arXiv preprint arXiv:2502.18387*.

Minhua Lin, Zongyu Wu, Zhichao Xu, Hui Liu, Xianfeng Tang, Qi He, Charu Aggarwal, Xiang Zhang, and Suhang Wang. 2025b. A comprehensive survey on reinforcement learning-based agentic search: Foundations, roles, optimizations, evaluations, and applications. *arXiv preprint arXiv:2510.16724*.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. 2024a. Time-mmd: A new multi-domain multimodal dataset for time series analysis. *arXiv preprint arXiv:2406.08627*.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. 2024b. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM on Web Conference 2024*, pages 4095–4106.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2022. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7628–7636.

Bhavitvya Malik, Abhinav Ramesh Kashyap, Min-Yen Kan, and Soujanya Poria. 2023. Udapter–efficient domain adaptation using adapters. *arXiv preprint arXiv:2302.03194*.

Jingchao Ni, Zhengzhang Chen, Wei Cheng, Bo Zong, Dongjin Song, Yanchi Liu, Xuchao Zhang, and Haifeng Chen. 2021. Interpreting convolutional sequence model by learning local prototypes with adaptation regularization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1366–1375.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Yiming Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. 2025. Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. *arXiv preprint arXiv:2503.07649*.

Jorge Nino, Andrés Frias-Velazquez, Nyan Bo Bo, Maarten Slembrouck, Junzhi Guan, Glen Debard, Bart Vanrumste, Tinne Tuytelaars, and Wilfried Philips. 2016. Scalable semi-automatic annotation for multi-camera person tracking. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Christopher Reining, Fernando Moya Rueda, Friedrich Niemann, Gernot A. Fink, and Michael ten Hompel. 2020. Annotation performance for multi-channel time series har dataset in logistics. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*.

ChengAo Shen, Zhengzhang Chen, Dongsheng Luo, Dongkuan Xu, Haifeng Chen, and Jingchao Ni. 2025. Exploring multi-modal data with tool-augmented llm agents for precise causal discovery. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 636–660.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. 2024. Test: Text prototype aligned embedding to activate llm's ability for time series. In *ICLR*.

Junxiang Wang, Guangji Bai, Wei Cheng, Zhengzhang Chen, Liang Zhao, and Haifeng Chen. 2024. Pond: Multi-source time series domain adaptation with information-aware prompt tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3140–3151.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

Junjie Xu, Zongyu Wu, Minhua Lin, Xiang Zhang, and Suhang Wang. 2024. Llm and gnn are complementary: Distilling llm for multimodal graph learning. *arXiv preprint arXiv:2406.01032*.

Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. 2021. Voice2series: Reprogramming acoustic models for time series classification. In *International conference on machine learning*, pages 11808–11819. PMLR.

Kristina Yordanova and Frank Krüger. 2018. Creating and exploring semantic annotation for behaviour analysis. *Sensors*.

Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. 2023. Temporal data meets llm–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *ICLR*.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355.

A More Related Work

A.1 LLMs for Time Series Analysis

The rapid advancement of LLMs in natural language processing has unveiled unprecedented capabilities in sequential modeling and pattern recognition, which can be leveraged for time series analysis. Three primary approaches are commonly adopted (Jiang et al., 2024): direct querying of LLMs (Xue and Salim, 2023; Yu et al., 2023; Gruver et al., 2024), fine-tuning LLMs with task-specific modifications (Chang et al., 2023; Cao et al., 2024; Jin et al., 2024; Sun et al., 2024), and incorporating LLMs into time series models to enhance feature extraction (Li et al., 2024).

Direct querying involves using LLMs to generate predictions or identify patterns from the data without modifying the underlying architecture. For example, PromptCast (Xue and Salim, 2023) applies LLMs to time series forecasting through a sentence-to-sentence paradigm. Yu et al. explore the use of LLMs for domain-specific tasks like financial time series forecasting (Yu et al., 2023), while LLMTime (Gruver et al., 2024) demonstrates how LLMs can function as effective learners by tokenizing time series data in a text-like format.

Fine-tuning LLMs enables them to better capture the intricacies of time series data by adapting them to specific datasets or tasks. For instance, LLM4TS (Chang et al., 2023) shows that fine-tuning pre-trained models can enhance forecasting performance. Additionally, TEMPO (Cao et al., 2024) and TEST (Sun et al., 2024) introduce architectures tailored for time series prediction, further demonstrating the power of specialized designs.

Lastly, LLMs can also act as feature enhancers within traditional time series models, enriching data representations and boosting performance. For example, (Li et al., 2024) illustrates how a frozen LLM can augment zero-shot learning for ECG time series analysis, highlighting the potential of LLMs to provide valuable features for complex datasets.

More recently, retrieval-augmented designs have been proposed for *zero-shot forecasting*. TS-RAG (Ning et al., 2025) retrieves semantically relevant time-series segments from a knowledge base and fuses them with a forecasting backbone to improve robustness and interpretability. In contrast, BRIDGE (Li et al., 2025a) focuses on text-controlled time-series generation, using natural-language descriptions and an LLM-based multi-agent pipeline to synthesize paired text–time-series data. Unlike these forecasting- or generation-oriented approaches, our work targets cross-domain time-series annotation, aiming to produce general and domain-specific semantic annotations that support interpretability and downstream tasks.

A.2 Domain Specialization of LLMs

Domain specialization of LLMs refers to the process of adapting broadly trained models to achieve optimal performance within a specific domain. This is generally categorized into three approaches: prompt crafting (Ben-David et al., 2022; Zhang et al., 2023; Xu et al., 2024; Lin et al., 2025a), external augmentation (Izacard et al., 2023), and model fine-tuning (Malik et al., 2023; Pfeiffer et al., 2020; Lin et al., 2025b). One of the earliest efforts in this area is PADA (Ben-David et al., 2022), which enhances LLMs for unseen domains by generating domain-specific features from test queries and using them as prompts for task prediction. Auto-CoT (Zhang et al., 2023) advances domain specialization by prompting LLMs with the phrase "Let's think step by step," helping guide the models in generating reasoning chains. Additionally, Izacard et al. (2023) propose integrating a relatively lightweight LLM with an external knowledge base, achieving performance comparable to much larger models like PaLM (Chowdhery et al., 2023). These studies highlight the flexibility of LLMs in adapting to specific domains through various strategies for domain adaptation.

A.3 Cross-modality Knowledge Transfer Learning through Pre-trained Models

There has been growing interest in leveraging pre-trained models for cross-modality knowledge transfer, particularly between the language, vision, and time series domains (Bao et al., 2022; Lu et al., 2022; Yang et al., 2021; Zhou et al., 2023). For instance, Bao et al. (2022) proposes a stagewise pre-training strategy that trains a language expert using frozen attention blocks pre-trained on image-only data. Similarly, Lu et al. (2022) examines the transferability of language models to other domains, while Zhou et al. (2023) applies pre-trained language and image models to time series analysis tasks. To the best of our knowledge, no previous work has specifically explored cross-modality knowledge transfer for time series annotation. Our work aims to fill this gap by investigating the application of cross-modality transfer learning in the context of automatic time series annotation.

B Notations

Table 6 presents all the notations we used in this paper.

Table 6: Notation Table

Symbol	Description
$\mathbf{x}$	Input time series data
e_s	Domain-specific annotation from source domains
e_t	Domain-specific annotation from target domain
e_d	Domain-decontextualized annotation
e_g	General annotation
f_t	Time-series-wise feature
f_l	Text-wise feature
$\mathcal{J}$	Domain-specific term (jargon) from target domain
m_d	Domain decontextualizer
m_t	Time-series-wise feature extractor
m_l	Text-wise feature extractor
m_{sel}	Feature selector
m_{gen}	General annotator
m_{jar}	domain-specific term extractor
m_{spe}	Domain-specific annotator
m_{rev}	Annotation reviewer
p_{de}	prompt of domain-decontextualization
p_l	prompt of text-wise feature extraction
p_{score}	prompt of scoring
p_{gen}	prompt of general annotation
p_{ext}	prompt of domain-specific term extraction
p_{spe}	prompt of domain-specific annotation
p_{rev}	prompt of annotation review

Additionally, we also provide some specific examples of domains, annotations, and features to improve the clarity of the problem settings of cross-domain time series annotation defined in Section 3. Specifically, our paper consider six distinct domains, i.e., Stock, Health, Environment, Social Good, Climate and Economy. For instance, the stock dataset includes multivariate time series data (e.g., stock price, volume, RSI, moving average) with corresponding annotations capturing features like support levels and resilience. More examples of text-wise features can be found in Table 19. More examples of domain-specific annotations are provided in Tables 33, 34, 35, 36, 37 and 38 of Appendix M.‘

C More Details of Multi-modal Feature Extraction

C.1 Time-series Feature Extraction

Given a time series data $\mathbf{X} = \{(\mathbf{x}_1, \cdots, \mathbf{x}_L)\}$, we develop a time series extraction toolbox $\{f_t^1, \ldots, f_t^{N_t}\}$ to extract time-series-wise features from $\mathbf{X}$. Specifically, we include *seasonality*, *trend*, *noise*, *moving average*, *lag feature*, *rolling window feature*, and *Fourier frequency* as intra-variable time-series-wise features. For multivariate time series, we also consider inter-variable time-series-wise features, i.e., *mutual information*, *Pearson correlation*, and *canonical correlation*.

In particular, we employ Seasonal-Trend decomposition (STL) (Cleveland et al., 1990) to extract seasonality, trend, and noise from the given time series data. To extract Fourier frequencies, the Fast Fourier Transform (FFT) (Almeida, 1994) is applied to convert a time-domain signal into its frequency components. For the inter-variable time-series features, we use `np.corrcoef` to compute the Pearson correlation. To calculate *mutual information*, two time series are first discretized, followed by `sklearn.metrics.mutual_info_score`. To calculate *canonical correlation*, we first use `sklearn.cross_decomposition` to decompose two time series data, and then use `np.corrcoef` to obtain the correlation.

C.2 Domain Decontextualization

We present the prompt template for domain decontextualization in Fig. 13.

C.3 Text Feature Extraction

Table 39 shows the prompt template for textual feature extraction.

D More Details of Adaptive Feature Selection

D.1 Offline LLM-based Feature Selection

The templates for p_{score} in Eq. (4) are shown in Fig. 14 and Fig. 15, respectively.

In some cases, we cannot input all the annotations to LLMs for calculating scores. We may split the annotations into several small batches and input the annotations in the small batches to calculate the score using Eq. (4). After that, we will accumulate the scores from all batches to get the final scores of each feature/token and then select the features with the top-k highest scores.

D.2 Incremental Reinforcement Learning-based Feature Selection

The necessity of this component. In the proposed LLM-based feature selection from Section 3, when new annotations exhibit different distributions or feature characteristics compared to the old data, it becomes necessary to re-query both old and new data to select the top-k most important features. This process is computationally intensive and resource-inefficient, especially as the volume of data grows. To address this issue, we propose the incremental reinforcement learning (RL)-based feature selection method. This approach provides the following benefits:

- **Cost-Efficiency**: Instead of re-querying LLMs with all the data, the RL-based method incrementally updates the knowledge stored in policy networks, requiring only the new data during updates.

- **Scalability**: By reducing redundant computations, the incremental RL-based method ensures scalability in dynamic environments with evolving data.

E Time Complexity Analysis

To analyze the time complexity of TESSA, we consider each component of TESSA separately, focusing on the computational cost associated with feature extraction, feature selection, and annotation generation and review.

Feature Extraction. Extracting intra-variable and inter-variable features has a complexity of $\mathcal{O}(C^2 \cdot n)$, where C is the number of channels in the time series and n is the number of time points.

Feature Selection. For offline LLM-based feature selection, the complexity is $\mathcal{O}(k \cdot M \cdot L^2)$, where k is the number of features, M is the model size (number of parameters) of the LLM, and L is the input sequence length. Incremental RL-based selection reduces this overhead by incrementally updating policy networks without re-querying old data.

Annotation Generation and Review. Each LLM inference for annotation generation or review has a complexity of $\mathcal{O}(M \cdot L^2)$. Given T samples, the overall complexity becomes $\mathcal{O}(T \cdot M \cdot L^2)$.

Overall Complexity. The combined complexity can be expressed as:

$$\mathcal{O}(T \cdot [C^2 \cdot n + k \cdot M \cdot L^2]),$$

where T is the number of time series samples.

F Experimental Settings

F.1 Dataset Statistics

Datasets. To evaluate the effectiveness of TESSA, five real-world datasets from distinct domains are considered: Stock, Health, Energy, Environment, and Social Good. Specifically, the stock dataset includes 1,935 US stocks with the recent 6-year data, collected from Investtech[2]. The other four datasets come from the public benchmark Time-MMD (Liu et al., 2024a). The dataset statistics are summarized in Table 7.

Additionally, we generate a synthetic dataset containing both time series and ground-truth annotations to directly assess the quality of the general annotations. The synthetic dataset is created by combining several key components from the time-series data:

- **Trend**: Introduces an overall direction, which can be upward, downward, or mixed.

- **Seasonality**: Adds cyclical patterns, modeled using sine waves.

- **Fourier Feature**: Incorporates complex periodic behavior by combining multiple sine and cosine waves.

- **Noise**: Adds Gaussian noise to simulate random fluctuations and real-world imperfections.

- **Rolling Window Features**: Captures smoothed trends (mean) and local variability (max/min).

- **Lag Features**: Uses past values to capture autocorrelation in the time series.

Ground-truth annotations are then generated by summarizing the key components of the synthetic time series.

In our synthetic dataset, we conduct 100 times random generation of each components and then combine them together to get 100 synthetic time series data, each with corresponding textual annotation.

F.2 Baseline Methods

Three baselines are applied in our general annotation evaluation for downstream tasks:

- **No Text**: No textual data are utilized in the forecasting process.

[2]https://www.investtech.com/

Table 7: Dataset Statistics

Domain	Frequency	# Channels	# Timestamps	# Samples
Stock	Daily	4	854,878	1,758
Health	Weekly	1	1,389	1,356
Social Good	Monthly	1	916	497
Energy	Daily	1	1,622	1,586
Environment	Daily	1	11,102	1,935
Climate	Monthly	5	496	177
Economy	Monthly	3	423	410

- **Time-MMD** (Liu et al., 2024a): A multimodal benchmark for time series analysis that incorporates both time series and text data. To adapt this method to our setting, we apply the original text data from the target datasets in (Liu et al., 2024a) to the forecasting task.

- **DirectLLM**: Directly uses the annotations generated by LLMs for time series forecasting. In this paper, we compare several representative LLMs in our evaluations.

F.3 Framework for Multi-modal Downstream tasks

To evaluate the quality of general annotations, we leverage the multi-modal time series analysis framework proposed in Time-MMD (Liu et al., 2024a), illustrated in Fig. 4. Using time series forecasting as a representative task, this framework employs an end-to-end pipeline that combines open-source language models with diverse time-series forecasting (TSF) models. Time-series data and textual annotations are modeled independently through dedicated unimodal TSF architectures and language models (LLMs) equipped with projection layers. The outputs of these modalities are fused via a dynamic linear weighting mechanism to generate final predictions. To optimize computational efficiency, we keep the LLM parameters frozen during training and update only the projection layers. Additionally, pooling layers are introduced to resolve dimension mismatches between textual variables and time-series features. The framework supports end-to-end training with minimal parameter overhead, ensuring both scalability and practicality.

F.4 Comparing TSSA with End-to-End Single-modal Fine-tuning

To further validate the effectiveness of TESSA, we implement another baseline that jointly fine-tuning one LLM on both source and target domain datasets. Specifically, we fine-tune Qwen2-7B-Instruct (via LoRA) on source (Stock, Health) and target (SocialGood) datasets. The time series forecasting results and the LLM-as-a-Judge results are provided in Table 8 and Table 9, respectively. From these tables, we observe that (i)the fine-tuning baseline slightly outperforms DirectLLM in time series forecasting, but largely underperforms TESSA across both MSE and MAE; (ii) The fine-tuning baseline improves domain relevance slightly over DirectLLM, but it does not improve clarity or comprehensiveness and still underperforms TESSA across all metrics. This supports our design hypothesis: monolithic models struggle to balance general and domain-specific reasoning, while our modular approach enables more effective and specialized annotation.

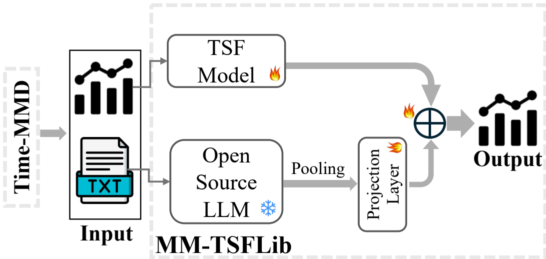


Figure 4: Overall framework of MM-TSFlib from Time-MMD (Liu et al., 2024a) used in our multi-modal downstream tasks. MMTSFlib uses a model-agnostic multi-modal integration framework that independently models time-series and textual annotations within an end-to-end training manner. MM-TSFlib slightly increases the number of trainable parameters, balancing effectiveness and efficiency.

F.5 Comparison with Existing Multi-agent Systems

To better contextualize TESSA's contributions against general-purpose multi-agent frameworks, such as AutoGen (Wu et al., 2024), MetaGPT (Hong et al., 2024), CAMEL (Li et al., 2023), we argue that while general-purpose

Table 8: Forecasting results with Qwen2-7B-Instruct as the LLM backbone. Informer is the time series forecasting model. SocialGood is the target dataset. Stock and Health are the source datasets. **Finetuned** is the baseline jointly fine-tuning the LLM on both source and target domain datasets.

Metric	NoText	TimeMMD	DirectLLM	Finetuned	TESSA
MSE	2.1457	1.6072	0.5550	0.4612	**0.3651**
MAE	1.1205	0.9731	0.4850	0.3792	**0.2838**

Table 9: Domain-specific annotation results on the Social Good dataset with Qwen2-7B as the LLM backbone. SocialGood is the target dataset. Stock and Health are the source datasets. **Finetuned** is the baseline jointly fine-tuning the LLM on both source and target domain datasets.

Metric	DirectLLM	Finetuned	TESSA
Clarity	3.28	3.23	**4.68**
Compre.	3.26	3.18	**4.49**
Dom. Rel.	3.33	3.53	**4.45**
Overall	3.29	3.33	**4.48**

systems like AutoGen, MetaGPT, and CAMEL have shown promise in task planning and code generation, **they are not tailored for time series annotation**.

To evaluate their suitability, we conducted a new experiment by adapting AutoGen for our task. We defined three roles:

- A **general annotator** to produce initial time series descriptions,
- A **domain-specific annotator** to specialize these with in-domain terminology, and
- An **annotation reviewer** to provide iterative feedback.

We configured AutoGen to use the same LLM backbone and prompt templates as TESSA to ensure a fair comparison. Following the setup in Section 4.4, we evaluated all models on the SocialGood dataset using our LLM-as-a-Judge framework. Results are reported in Table 10. These results show that while AutoGen improves over DirectLLM, it does not match TESSA's performance. We believe this is due to key architectural differences. AutoGen supports sequential agent coordination but lacks explicit mechanisms for (i) cross-domain feature identification, and (ii) conditioning on domain-specific terminology extracted from limited target-domain annotations.

In contrast, TESSA incorporates:

- **LLM-guided feature selection**, which promotes the extraction of robust general patterns across domains (Sec. 3.1),
- A **Domain-Specific Term Extractor**, which identifies key jargon from sparse annotations (Sec. 3.3).
- A tightly integrated **Annotation Reviewer**, which forms a bi-directional loop that iteratively refines annotation quality (Sec. 3.4).

This comparison highlights that careful design of agent roles and domain adaptation mechanisms is essential for time series annotation.

Table 10: Comparison between TESSA and AutoGen on domain-specific annotation. The target dataset is SocialGood, with Qwen2-7B as the LLM backbone. Stock and Health serve as the source datasets.

Metric	DirectLLM	AutoGen	TESSA
Clarity	3.28	3.89	**4.68**
Compre.	3.26	3.93	**4.49**
Dom. Rel.	3.33	4.01	**4.45**
Overall	3.29	3.94	**4.48**

F.6 Additional Experiments on Traditional Time Series Data

In this subsection, we adopt TESSA to a widely used dataset ETT-small_h1 (Zhou et al., 2021), which is a traditional time series dataset without accompanying textual inputs. To adopt TESSA to this setting, we use OpenAI's o4-mini to generate domain-relevant terminology that represents the kind of specialized language that would typically accompany this dataset. These generated domain-specific terms are then fed into the Domain-Specific Annotator to guide annotation. Following the evaluation setup in Section 4.4, we compare TESSA with DirectLLM in generating domain-specific annotations. The results are summarized in Table 11. These results demonstrate that TESSA can still generate **high-quality time series annotations** on traditional datasets, even without explicit textual inputs, outperforming DirectLLM across all metrics.

Table 11: Comparison results on the ETT-small_h1 target dataset with GPT-4o as the LLM backbone. Stock and Health are the source datasets.

Metric	DirectLLM	TESSA
Clarity	3.32	**4.69**
Compre.	3.45	**4.57**
Dom. Rel.	3.07	**4.54**
Overall	3.28	**4.59**

F.7 Comparison with More Traditional Time Series Models

In this subsection, to further enrinch the evaluation, in addition to Informer, we add two more strong traditional baselines (i.e., Reformer (Kitaev et al., 2020) and PatchTST (Nie et al., 2023)) for the Energy dataset. The restuls are reported in Table 12. These results show that TESSA consistently improves forecasting performance across different backbone models. This highlights its ability to generate **semantically meaningful annotations** that benefit downstream time series tasks, even when integrated with traditional models.

Table 12: Forecasting results with GPT-4o as the LLM backbone. Energy is the target dataset. Stock and Health are the source datasets.

Backbone	Metric	NoText	DirectLLM	TESSA
Reformer	MSE	0.0305	0.0287	0.0238
	MAE	0.1234	0.1198	0.1100
PatchTST	MSE	0.2440	0.1432	0.0942
	MAE	0.2321	0.2104	0.1844

G Additional Results for General Annotation Evaluation in Downstream Tasks

G.1 Implementation Details

Time Series Forecasting Models. We use Informer (Zhou et al., 2021) as the forecasting model for the time series forecasting task. The model is configured with a dropout rate of 0.1 and a learning rate of 0.0001.

Large Language Models. We utilize GPT-4o (Achiam et al., 2023), along with two open-source models: LLaMA3.1-8B (Dubey et al., 2024) and Qwen2-7B (Yang et al., 2024). For the open-source models, we set `temperature=1` and `max_tokens=2048`, while all other settings follow the defaults.

Each experiment in our paper is conducted five times, with the average result reported. All models are trained on an Nvidia A6000 GPU with 48GB of memory.

G.2 Evaluation in Time Series Forecasting Tasks

The full results are presented in Table 13. From the table, we can observe the following: (1) TESSA consistently outperforms all baselines across all settings, demonstrating its effectiveness in generating high-quality general annotations. (2) Among the three LLMs, GPT-4o-backed TESSA achieves the best performance, outperforming both LLaMA3.1-8B and Qwen2-7B. We attribute this to the higher quality of the annotations generated by GPT-4o compared to the other models, further emphasizing that high-quality annotations can significantly enhance downstream task performance.

G.3 Evaluation in Time Series Imputation Tasks

To demonstrate the effectiveness of TESSA in improving the performance of various downstream tasks, we further apply the generated general annotations in time series imputation task. Specifically, time series imputation task refers to the process of filling in missing or incomplete data points in a time series dataset, where some values are randomly mask.

Implementation Details. We implement the multi-modal time series imputation based on TSLib (Wu et al., 2023). We use Informer (Zhou et al., 2021) as the forecasting model for the time series forecasting task. The model is configured with a dropout rate of 0.1 and a learning rate of 0.0001. GPT-4o is set as the LLM backbone. Other settings follow these in Section G.1.

Experimental Results. The experimental results are shown in Table 14. From the table, we observe that TESSA consistently outperforms baselines in all datasets, demonstrating that TESSA's annotations can significantly benefits various downstream tasks, including forecasting and imputation.

Table 13: Comparison results in forecasting. Informer is the time series forecasting model.

Domain	Backbone	Metrics	No Text	Time-MMD	DirectLLM	TESSA
Environment	GPT-4o	MSE	1.2542	0.8483	0.7714	**0.4629**
		MAE	0.7387	0.6865	0.6604	**0.4424**
	LLaMA3.1-8B	MSE	1.2542	0.8483	0.8108	**0.5654**
		MAE	0.7387	0.6865	0.6805	**0.5128**
	Qwen2-7B	MSE	1.2542	0.8483	0.7956	**0.5824**
		MAE	0.7387	0.6865	0.6729	**0.5419**
Energy	GPT-4o	MSE	2.0117	0.2172	0.0575	**0.0482**
		MAE	1.1663	0.2139	0.0055	**0.0040**
	LLaMA3.1-8B	MSE	2.0117	0.2172	0.1023	**0.0531**
		MAE	1.1663	0.2139	0.0130	**0.0049**
	Qwen2-7B	MSE	2.0117	0.2172	0.0824	**0.0522**
		MAE	1.1663	0.2139	0.0097	**0.0048**
Social Good	GPT-4o	MSE	2.1457	1.6072	0.4639	**0.1935**
		MAE	1.1205	0.9731	0.3801	**0.0825**
	LLaMA3.1-8B	MSE	2.1457	1.6072	0.6720	**0.3422**
		MAE	1.1205	0.9731	0.6138	**0.2489**
	Qwen2-7B	MSE	2.1457	1.6072	0.5550	**0.3651**
		MAE	1.1205	0.9731	0.4850	**0.2838**

Table 14: Imputation results with GPT-4o as the LLM backbone. Informer is the imputation model.

Metric	Domain	NoText	TimeMMD	DirectLLM	TESSA
MSE	Environment	0.9718	0.9657	0.9453	**0.5698**
	Energy	0.9109	0.9081	0.9018	**0.8690**
	Social Good	1.4971	0.9784	0.6873	**0.5492**
MAE	Environment	0.6872	0.6867	0.6973	**0.5438**
	Energy	0.8216	0.8176	0.8111	**0.8075**
	Social Good	0.8371	0.7806	0.6036	**0.5116**

H Additional Details of General Annotation Evaluation in Synthetic Datasets

H.1 Implementation Details

To evaluate the effectiveness of TESSA in generating general annotations for synthetic time series using an LLM-as-a-judger approach, we set GPT-4o as the backbone of the judger. Two metrics, *Clarity* and *Comprehensiveness*, are used to assess the quality of the annotations:

- **Clarity**: Evaluates the clarity and readability of the annotations.
- **Comprehensiveness**: Assesses whether the annotations cover the most important patterns.

H.2 Prompt Templates of LLM-as-a-judger

The prompts for evaluations are shown in Table 40 and Table 41.

I Additional Results of Domain-specific Annotation Evaluation

I.1 Evaluation Metrics

We use the following three metrics to evaluate the quality of domain-specific annotations:

- **Clarity**: Assesses the clarity and readability of the annotations.
- **Comprehensiveness**: Checks whether the annotations cover the most important patterns.
- **Domain-Relevance**: Evaluates whether the annotations correctly apply domain-specific knowledge.

I.2 Prompt Template

The prompts used to evaluate the domain-specific annotations based on the three metrics are shown in Table 42, Table 43, and Table 44, respectively.

I.3 Additional Results on Other LLM Backbones

We report the evaluation results of the domain-specific annotations on the Energy and Social Good datasets in Table 15 and Table 16, respectively. Similar observations are made in Section 4.4, further demonstrating the effectiveness of TESSA in generating high-quality domain-specific annotations.

Table 15: Domain-specific annotation results on the Energy dataset with GPT-4o as the LLM backbone.

Metric	Method	Mean	P(T>D) (%)
Clarity	TESSA DirectLLM	**4.79** 3.48	**99.35**
Compre.	TESSA DirectLLM	**4.57** 3.10	**98.01**
Dom. Rel.	TESSA DirectLLM	**4.25** 3.01	**95.24**
Overall	TESSA DirectLLM	**4.57** 3.35	**98.31**

Table 16: Domain-specific annotation results on the Social Good dataset with GPT-4o as the LLM backbone.

Metric	Method	Mean	P(T>D) (%)
Clarity	TESSA DirectLLM	**4.68** 3.28	**99.61**
Compre.	TESSA DirectLLM	**4.49** 3.26	**97.54**
Dom. Rel.	TESSA DirectLLM	**4.45** 3.33	**95.34**
Overall	TESSA DirectLLM	**4.48** 3.29	**97.16**

J Experimental results of Human in Loop

In this section, we demonstrate that the annotations generated by TESSA can assist humans in analyzing time-series data. To evaluate this, we selected 60 time-series samples from three domains, Environment, Energy, and Social Good, with 20 datasets per domain. We compared annotations generated by TESSA and DirectLLM, asking 20 PhD stduents, researchers, and professors as the participants to assess which annotations were more informative and useful. The results of this human-in-the-loop evaluation, summarized in Table 17, reveal that TESSA consistently outperforms DirectLLM across all three domains. Participant assessments indicate that 88.3% of TESSA's general annotations and 93.3% of its domain-specific annotations are more informative compared to DirectLLM's outputs. This substantiates TESSA 's capacity to produce semantically meaningful annotations that enhance human interpretability during time-series analysis workflows.

Table 17: Comparison results of general and domain-specific annotations. GPT-4o is the LLM backbone.

Domain	Method	P(T>D) (%)
Environment	General Specific	83.3 91.7
Energy	General Specific	88.3 93.3
Social Good	General Specific	85.8 92.5

K Additional Results on Multivariate Time Series

In this section, we aim to demonstrate the effectiveness of TESSA in generalizing to multivariate time series. Specifically, We use Stock and Health datasets as the source domains, Climate and Economy datasets as the target domains.

Domain-specific Annotations Evaluation. Similar to Section 4.4, we adopt a LLM-as-a-Judger strategy to evaluate the performance of TESSA and DirectLLM in generating domain-specific annotations. Other settings follow these in Section 4.4. We present the comparison results on the two datasets in Table 18. From the table, we observe that similar to these of univariate time series in Section 4.4, TESSA significantly outperforms DirectLLM across all metrics in both two multivariate time series datasets, aciving an overall score of 4.51 and 4.55 compared to DirectLLM's 3.38 and 3.42 in Climate and Economy dataset, respectively. This demonstrates the effectiveness of TESSA in generalizing to multivariate time series. Additional case studies of TESSA applied to multivariate time series are presented in Appendix M.

L Additional Details of Ablation Studies

L.1 Evaluation Metric

To evaluate the effectiveness of the adaptive feature selection, we use an LLM-as-a-judger to evaluate the general annotations generated by TESSA and its variant TESSA/F. We propose to evaluate it by using the prompt in Table 45.

Table 18: Comparison results of domain-specific annotations for multivariate time series in the Climate and Economy datasets using GPT-4o as the LLM backbone.

Domains	Metric	Method	Mean	P(T>D) (%)
Climate	Clarity	TESSA	**4.58**	**99.17**
		DirectLLM	3.35	
	Compre.	TESSA	**4.37**	**96.43**
		DirectLLM	3.38	
	Dom. Rel.	TESSA	**4.57**	**96.20**
		DirectLLM	3.40	
	Overall	TESSA	**4.51**	**97.26**
		DirectLLM	3.38	
Economy	Clarity	TESSA	**4.46**	**97.14**
		DirectLLM	3.48	
	Compre.	TESSA	**4.57**	**96.81**
		DirectLLM	3.41	
	Dom. Rel.	TESSA	**4.61**	**96.70**
		DirectLLM	3.37	
	Overall	TESSA	**4.55**	**96.88**
		DirectLLM	3.42	

Table 19: Ablation studies of the impact of domain decontextualization. Red denotes the irrelevant features.

TESSA's extracted text-wise features: support level, resistance level, volume correlation, breakthrough, trend reversal, relative strength index, negative signal, positive signal, channel boundaries
TESSA/D's extracted text-wise features: higher prices over time, autocorrelation, price increase, trend channel, stationary, fun, lower prices, outliers, breakdown, rising trend

L.2 Qualitative Examples

We present a qualitative example of extracting text-wise features using TESSA and TESSA/D, shown in Table 19. From the table, we observe that TESSA/D captures irrelevant features, such as *higher prices over time* and *fun*, which are unrelated to time series analysis. This supports our claim that domain-specific terminology can hinder the accurate extraction of time-series-relevant features.

We also provide another qualitative example in Table 21 to demonstrate the effectiveness of adaptive feature selection. The annotations generated by TESSA/F tend to include numerous features without proper analysis. This illustrates that including too many features can reduce the clarity of the annotations, further emphasizing the importance of adaptive feature selection in improving annotation quality.

Table 20: Case study: Evaluation results of domain-specific annotation of time series data in Fig. 6 from the Environment dataset. GPT-4o is the LLM backbone.

Metric	Method	Score
Clarity	TESSA	**5.0**
	DirectLLM	3.0
Compre.	TESSA	**3.0**
	DirectLLM	3.0
Dom. Rel.	TESSA	**5.0**
	DirectLLM	3.0
Overall	TESSA	**4.3**
	DirectLLM	3.0

L.3 Additional Ablation Studies on the Contributions of TESSA's Components

In Section 4.5, we conduct ablation studies to assess the importance of domain decontextualization and adaptive feature selection in TESSA. To further understand the contributions of TESSA's components, we conduct additional ablation studies on domain-specific term extractor and annotation reviewers, respectively.

Domain-specific Term Extractor. To understand the ability of LLMs to process domain-specific jargon to generate useful features, we implement a variant TESSA/S, which removes the domain-specific term extractor in TESSA and directly generate domains-specific annotations from general annotations. We provide the comparison results in Stock and SocialGood datasets in Tables 22 and 23, which demonstrate that TESSA is able to capture more jargon-rich in information, such as resistance levels in the Stock dataset and reactive bounds in the SocialGood dataset. In contrast, TESSA/S merely converts general features into basic domain-specific features. This highlights that LLMs, when used as domain-specific term extractors, can effectively generate valuable jargon that enhances the generation of domain-specific annotations.

Annotation Reviewer. To evaluate the impact of the annotation reviewer component, we introduce a variant TESSA/A that excludes this module. We assess the quality of domain-specific annotations generated by TESSA and TESSA/A using an LLM-as-a-judger framework, focusing on two criteria: (1) *Clarity* (readability and coherence of annotations) and (2) *Domain-Relevance* (alignment with domain-specific context). An overall score, calculated as the average of these metrics, provides a holistic performance measure. As shown in Table 24, TESSA achieves statistically significant

Table 21: Ablation studies of the impact of adaptive feature selection. Two compared methods are used for generating general annotation ploted in Fig. 6(a). Purple highlights basic patterns, green highlights general patterns.

TESSA's generated general annotation: The time series data tracks a single variable over time, showing a gradual upward trend from around 6.96 to over 10.4. There are clear seasonal patterns, with regular cycles of peaks and troughs, hinting at periodic influences like seasonal demand. The data exhibits strong support and resistance levels, where it hesitates at specific values before continuing its rise, demonstrating resilience. There is a consistent upward indicator trend, suggesting stable or improving underlying factors driving this increase. Overall, the data indicates a positive growth trajectory, with predictable cyclical fluctuations due to natural or recurring factors.
TESSA/F's generated general annotations: The provided time series data shows a clear upward trend, indicated by the gradual increase in trend values over time. Initially, fluctuations are minor, but as the series progresses, we observe larger increases, especially towards the end of the sequence, suggesting a strong positive momentum. The occasional deviations from the trend, as shown by the residue, hint at short-term fluctuations or potential anomalies. Seasonality seems to have minimal impact with slight cyclical patterns, possibly reflecting periodic but non-dominant effects. Overall, if this time series pertains to a scenario where higher values are desirable, such as sales or economic indicators, this upward trend and increased values could be considered as positive signals or a breakthrough.

improvements over TESSA/A across all metrics. This quantitative comparison demonstrates that the annotation reviewer critically enhances both the clarity and contextual relevance of generated annotations, ensuring they better capture domain-specific nuances.

L.4 Additional Ablation Studies of Data Contamination

To investigate whether TESSA retains its effectiveness when the LLMs used are not exposed to data describing the time series, we conduct an ablation study using fully open LLMs, specifically OLMo-7B (Groeneveld et al., 2024). The categories of the training data for the LLM are listed in Table 25. In this study, we carefully select time series with textual annotations that neither appear in the training data of OLMo-7B nor describe the time series themselves. We use the Stock and Health datasets as source domains, and the Energy dataset as the target domain. We then apply TESSA and DirectLLM to generate both general and domain-specific annotations for the time series data in the Energy dataset.

The results are presented in Table 26. From the table, we observe that: (1) despite using a fully open LLM that has not been exposed to data describing the time series, TESSA with OLMo-7B as the LLM backbone is still able to understand and generate informative annotations with richer general and domain-specific patterns, using more natural language. In contrast, DirectLLM only offers a simplistic description of the basic trend of the time series data. This further underscores the effectiveness of TESSA even in a scenario of strict data non-contamination.

Additionally, we employ an LLM-as-a-Judger strategy to evaluate the domain-specific annotations generated by TESSA and DirectLLM, as shown in Table 26. The evaluation is conducted from three perspectives: Clarity, Comprehensiveness, and Domain-relevance. GPT-4o serves as the LLM judger, and the other settings follow those in Sec. 4.4. The comparison results are presented in Table 27. We observe that TESSA consistently achieves full scores in all three metrics, significantly outperforming DirectLLM. This further demonstrates TESSA 's effectiveness in generating high-quality domain-specific annotations with stronger clarity, comprehensiveness, and domain-relevant contextual precision.

M Additional Details of Case Studies

More Details of the Case Studies in Section 4.6 In this section, we provide additional details of the case studies in Section 4.6. We select a representative time series from the Social Good domain, shown in Fig. 6(b). In Table 28, both the general and domain-specific annotations generated by DirectLLM and TESSA are reported. We also quantitatively evaluate the domain-specific annotations of TESSA and DirectLLM, following the setup

Table 22: Ablation studies of the impact of the domain-specific term extractor in Stock dataset. Purple highlights basic patterns, green highlights general patterns, yellow highlights domain-specific patterns and blue highlights correlations between variables in multivariate time series data.

TESSA: Compass Digital Acquisition Unit's stock price shows a notable pattern of rising and falling periodically, indicating seasonality with stable long-term trends interrupted by short-term fluctuations. There are key resistance levels around intervals 134,270, and 403, where prices peak before dipping. The stock volume demonstrates significant spikes at specific points, suggesting irregular activity, particularly around values 7,000 and 80,500, which may indicate volume bursts or unusual market events. The relative strength index (RSI) also reveals a recurring pattern, gradually trending upward before a sharp decline, reflecting a cycle of growth and subsequent drop. Overall, the mild positive correlation between stock price and RSI indicates that periodic changes in price are somewhat echoed in RSI patterns, potentially offering predictive insights for future stock movements.
TEESA/S: Compass Digital Acquisition Unit's stock price exhibits a distinct cyclical pattern, suggesting seasonality, with occasional fluctuations. The relative strength index (RSI) shows a repeating trend, gradually rising before experiencing a sharp drop, indicating a pattern of growth followed by a decline. The mild positive relationship between stock price and RSI suggests that changes in price tend to be mirrored in the RSI, offering potential clues for predicting future price movements.

Table 23: Ablation studies of the impact of the domain-specific term extractor in SocialGood dataset. Purple highlights basic patterns, green highlights general patterns and yellow highlights domain-specific patterns.

TESSA: The unemployment rate displays a clear upward trend over time, starting from the mid-5s and progressing to the mid-7s toward the end of the series, signaling persistent growth in unemployment. This movement is influenced by volatility periods, suggesting that economic cycle phases are periodically impacting employment levels. The observed resistance thresholds around the mid-5s and mid-6s mark critical threshold levels, where unemployment temporarily stabilizes before continuing its upward momentum. Analyzing lag features, past values like 5.5 and 6.0 serve as baselines, helping to understand how unemployment has evolved over time. This steady upward movement, punctuated by cyclical variations, highlights the need for strategies that address both economic momentum indicators and short-term reactive bounds, ensuring a robust response to both long-term trends and periodic disruptions in the labor market.
TEESA/S: The unemployment rate data reveals a consistent upward trend over time, starting from 5.18 and rising to approximately 7.24. This suggests an increasing trend in unemployment, which might pose socio-economic challenges if it persists. The data also exhibits seasonal fluctuations, indicating periods of elevated unemployment that could align with specific economic or legislative cycles. These recurring ups and downs highlight how external factors might periodically impact the job market. Recognizing these cyclical variations can help policymakers craft effective strategies to mitigate the potential socio-economic risks of rising unemployment in the future.

outlined in Section 4.4. The evaluation results are presented in Table 20.

From the table, we observe that (1) TESSA's general annotations capture more meaningful patterns, enhancing user understanding and supporting downstream tasks, while DirectLLM only highlights basic trends; and (2) TESSA's domain-specific annotations consistently outperform those of DirectLLM across all metrics, providing clearer, more comprehensive, and contextually relevant insights. Specifically, TESSA's annotations are more fluent, more detailed and provide a richer analysis using domain-specific jargons, like *economic momentum* and *labor market resilience*, while the annotations of DirectLLM only simply analyze the trend of the unemployment rate, providing less insights.

Case Study for Multivariate Time Series We then conduct a case study to demonstrate the effectiveness of TESSA in generating high-quality annotations for multivariate time series data. Specifically, we set the Stock dataset as the target domain. Health and Environment datasets are then applied in the source domains. The example multivariate time series data is shown in Fig. 8, where the multivariate time series data has four variables, i.e., price, volume, relative strength index (RSI) and simple moving average (SMA). The generated annotations are shown in Table 29. From the table, we observe that (1) TESSA's generated annotations are more natural than DirectLLM; (2) DirectLLM interprets each variable independently by only focusing their trends. However, TESSA can capture the correlation between variables. This shows TESSA is able to analyze inter-variable patterns. These further imply the effectiveness of TESSA in generating

Table 24: Ablation studies for the annotation reviewer in the Social Good dataset with GPT-4o as the LLM backbone.

Metric	Method	P(TESSA>TESSA/A) (%)
Clarity	TESSA TESSA/A	**95.4**
Dom. Rel.	TESSA TESSA/A	**87.6**
Overall	TESSA TESSA/A	**91.6**

Table 25: The category of the training data used to pre-train OLMo-7B (Groeneveld et al., 2024), which is from OLMo's technical report (Groeneveld et al., 2024).

Source	Type	UTF-8 bytes *(GB)*	Docs *(millions)*	Tokens *(billions)*
Common Crawl	web pages	9,812	3,734	2,180
GitHub	code	1,043	210	342
Reddit	social media	339	377	80
Semantic Scholar	papers	268	38.8	57
Project Gutenberg	books	20.4	0.056	5.2
Wikipedia	encyclopedic	16.2	6.2	3.7
Total		**11,519**	**4,367**	**2,668**

high-quality domain-specific annotations for multivariate time series data.

Case Study in the Synthetic Dataset. We further select an example from the synthetic dataset to conduct similar experiments to generate general annotations. The selected time series data is in Fig. 7. The qualitative example of the annotations of this time series data is shown in Table 30. From the table, we observe a discrepancy in DirectLLM's analysis, as it detects 138 values in the time series data, despite there being only 120 values. This leads to inaccurate annotations. Moreover, DirectLLM captures only the basic trend of the time series, whereas TESSA identifies more significant patterns, such as the *rolling window feature*, *seasonality*, and *resilience*. This demonstrates the effectiveness of TESSA in providing more comprehensive and accurate annotations. We analyze the reason TESSA mitigates the hallucination seen in DirectLLM is that it highlights important patterns overlooked by LLMs, such as seasonality. By focusing on these patterns rather than just basic trends, LLMs can analyze and interpret time series data from multiple perspectives, leading to fewer hallucinations in the annotations.

Additional Examples on Various Domains. Additional examples are presented for the synthetic,

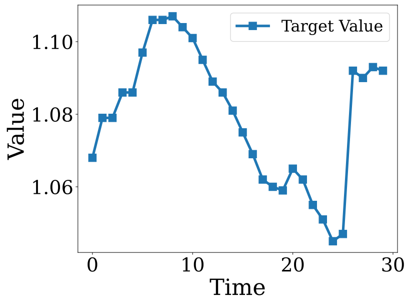
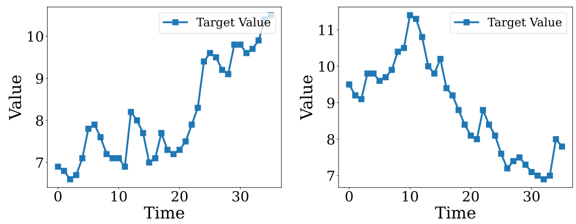


Figure 5: Selected time series data from Energy dataset for ablation studies on data continimation. This time series data has 30 data points. The corresponding generated annotations of TESSA and DirectLLMs are provided in Table 26.



(a) Ablation study (b) Case study

Figure 6: Selected time series data from Social Good dataset for ablation studies and case studies. (a) for ablation studies ; and (b) is for case studies.

environment, energy and social good datasets, respectively. Specifically, the general annotations of selected time series on the synthetic dataset that in Fig. 9 are shown in Tables 31 and 32. The domain-specific annotations of selected time series on the environment dataset (Fig. 10) are shown in Tables 33 and 34. And the domain-specific annotations of time series on the energy dataset (Fig. 11) are shown in Tables 35 and 36. Similarly, the domain-specific annotations of time series on the social good dataset (Fig. 12) are shown in Tables 37 and 38. Similar observations to those in Table 20 and Table 30 are found.

Table 26: Ablation studies on data contamination using OLMo-7B-Instruct. We select a time-series in the Energy dataset, which is shown in Fig. 5. Purple highlights basic patterns, green highlights general patterns, and yellow highlights domain-specific patterns.

TESSA's general annotations The time series data shows a trend of increasing and decreasing values with some seasonality , which can be observed from the values of the intra-variable time series features such as trend, seasonality , and moving average . The values of the intra-variable moving average are not stable, indicating that the time series data is volatile and prone to fluctuations . The values of the intra-variable lag features are not significant, indicating that there is no strong correlation among the time series data points. **DirectLLM's general annotations** This time series data represents a process with high variability in the values, with an increasing trend . The data is separated by '|' tokens, which could represent different stages or conditions of the process. The highest value of 1.108 is observed at the 11th instance, which might indicate a bottleneck or peak in the process. The series shows a decreasing trend from instances 9-12, which suggests that certain measures or interventions were put in place to address the bottleneck. Overall, the data suggests that the process is subject to fluctuations , but can be managed by identifying and addressing the root causes of the bottlenecks.
TESSA's domain-specific annotations The natural gas price is exhibiting a volatile pattern with significant fluctuations, suggesting a lack of stability in the market. This volatility hints at underlying market dynamics that are affecting price movements, potentially tied to factors such as supply and demand shifts or external economic variables . A closer examination of the trend channels indicates intermittent rising and falling trends , pointing towards a potential cyclical behavior in natural gas prices. Currently, the market does not show strong correlation from lag features, suggesting that recent price changes are not strongly influenced by past values. Support and resistance levels could play a crucial role, as breaking through these levels may signal a significant trend shift in the gas market's future pricing. **DirectLLM's domain-specific annotations** Gasonline prices in the Energy domain are fluctuating , ranging from 1.04 to 1.10 per unit, with occasional spikes above 1.10, and drops below 1.00. These prices fluctuate consecutively, indicating a dynamic market for Gasonline.

Table 27: Ablation studies of data contamination. We use an LLM-as-a-Judger to compare a domain-specific annotation in the Energy dataset generated by DirectLLM and TESSA. OLMo-7B is the LLM-backbone. GPT-4o is used as the LLM judger.

Metric	Method	Mean
Clarity	TESSA	**5**
	DirectLLM	3
Compre.	TESSA	**5**
	DirectLLM	2
Dom. Rel.	TESSA	**5**
	DirectLLM	3
Overall	TESSA	**5**
	DirectLLM	2.67

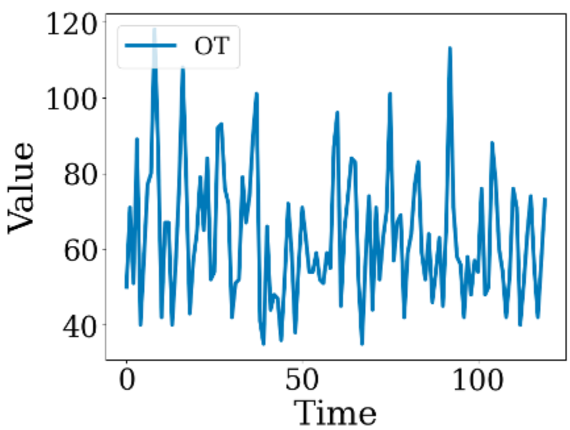


Figure 7: Case study: A selected time series data from the synthetic dataset. The time series data has 120 data points. OT denotes the target variable.

Table 28: A case study in Social Good dataset. The target time series data is shown in Fig. 6 (a). Purple highlights basic patterns, green highlights general patterns, and yellow highlights domain-specific patterns.

TESSA's general annotations

The time series data shows a generally decreasing trend, initially rising until a midpoint before consistently declining. Seasonal patterns are evident, with regular peaks around specific intervals. The series experiences rapid positive and negative fluctuations, reflecting significant changes in the measured system. Support and resistance levels are observed, with values frequently pausing near 7.0 and struggling to exceed 10.5. Past values have an impact on current data, indicating that recent patterns help predict future outcomes.

DirectLLM's general annotations

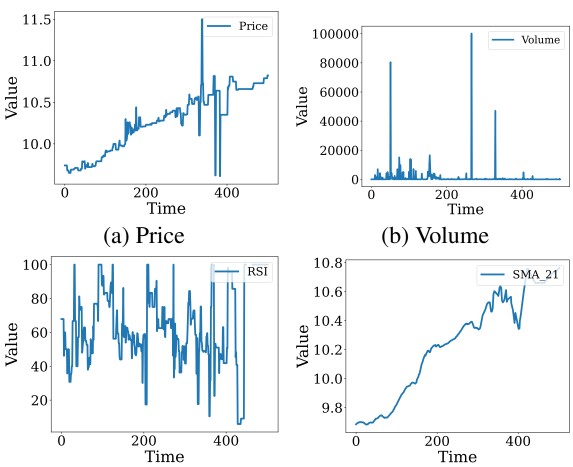
The uni-variate time series data exhibits fluctuating values with an overall descending trend. Initially, the data is relatively stable around the 9.5 to 10.5 range, peaking at 11.4 before descending gradually. Midway through the series, after reaching a peak, there is a noticeable decline as values drop from approximately 11 to levels around 8. By the end, the values stabilize somewhat but decline further, reaching a low near 7. This pattern suggests an initial increase followed by a downward trend and final stabilization at lower levels, indicating a potential cycle or shift in the underlying phenomenon being measured.

TESSA's domain-specific annotations

The unemployment rate shows a pattern consistent with trends in labor market resilience and economic momentum. Initially stable, with rates between 9 and 10, there is a gradual upward movement indicating early signs of economic growth and potential tightening in the labor market, peaking around 11. This is followed by a downturn, reflecting possible macroeconomic constraints or shifts that reduce economic momentum, as the rate drops to approximately 7.5 towards the end of the period. The rates around 9 and 11 can be seen as support and resistance thresholds for employment, where the market experiences inertia. Notably, the subtle seasonal fluctuations suggest the presence of moderate cyclic factors influencing employment trends, helping to understand broader economic activity patterns.

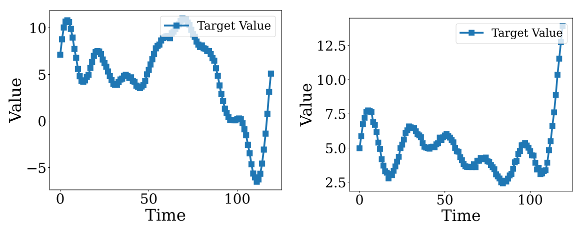
DirectLLM's domain-specific annotations

The time series data represents fluctuations in the unemployment rate over a specific period. Initially, the unemployment rate starts quite high, around 9.5%, and exhibits slight ups and downs but generally remains above 9% until a significant increase is observed, reaching its peak at 11.4%. Following this peak, the rate gradually declines, showing an overall downward trend with a notable reduction to as low as 7%. This decline suggests a potential improvement in the employment situation, perhaps due to economic recovery or effective employment policies. Towards the end of the series, the unemployment rate stabilizes around 7-8%, although a slight increase is seen again, indicating the dynamic nature of employment conditions.



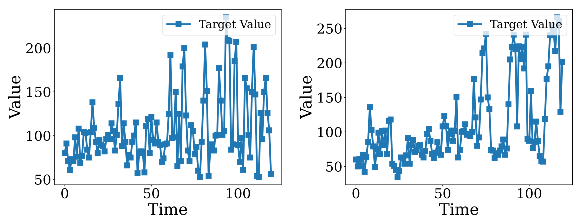
(a) Price (b) Volume (c) RSI (d) SMA

Figure 8: Case study: a multivariate time series data from the Stock dataset, which has four variables, i.e., price, volumn, RSI and SMA.



(a) (b)

Figure 9: More selected time series data from the synthetic dataset. The time series data has 120 data points.



(a) (b)

Figure 10: More selected time series data from the Environment dataset. The time series data has 120 data points.

Table 29: A case study in Stock dataset. The target **multivariate** time series data is shown in Fig. 8 (a). GPT-4o is the LLM backbone. Purple highlights basic patterns, green highlights general patterns, yellow highlights domain-specific patterns and blue highlights correlations between variables in multivariate time series data.

TESSA's domain-specific annotations
Compass Digital Acquisition Unit's stock price shows a notable pattern of rising and falling periodically, indicating seasonality with stable long-term trends interrupted by short-term fluctuations. There are key resistance levels around intervals 134, 270, and 403, where prices peak before dipping. The stock volume demonstrates significant spikes at specific points, suggesting irregular activity, particularly around values 7000 and 80500, which may indicate volume bursts or unusual market events. The relative strength index (RSI) also reveals a recurring pattern, gradually trending upward before a sharp decline, reflecting a cycle of growth and subsequent drop. Overall, the mild positive correlation between stock price and RSI indicates that periodic changes in price are somewhat echoed in RSI patterns, potentially offering predictive insights for future stock movements.

DirectLLM's domain-specific annotations
The provided time series data consists of three primary features: price, volume, and relative strength index (RSI). Over the observation period, the price demonstrates an overall upward trend, starting around \$9.74, exhibiting fluctuations, and rising to hover around \$10.81 towards the end. Notable price spikes correspond with significant increases in trading volume, indicating periods of high trading activity, such as jumps to 80,500 and 100,200 in volume. Additionally, the RSI values range sharply, highlighting areas of overbought conditions (RSI approaching or at 100) and oversold conditions (RSI dropping around or below 50). These RSI changes suggest periods of potential buying or selling pressure, mirroring the observed price moves.

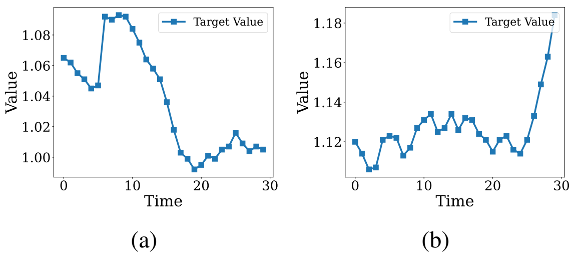


Figure 11: More selected time series data from the Energy dataset. The time series data has 36 data points.

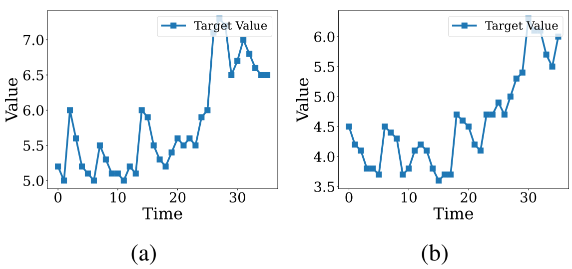


Figure 12: More selected time series data from the Social Good dataset. The time series data has 36 data points.

Table 30: A case study in the synthetic dataset. The selected time series data is shown in Fig. 7, where the time series data has 120 data points. Purple highlights basic patterns, green highlights general patterns, and red highlights hallucinations.

TESSA's general annotations This time series exhibits notable fluctuations with values ranging from 35.0 to 118.0. The trend indicates a general decline, starting around 65.26 and ending around 53.58 reflecting a significant downward movement over time. Periodic peaks are captured by the rolling window feature showing intermittent spikes up to 118.0, suggesting instances of strong breakout behavior despite an overall support decrease. The seasonality component reveals cyclical patterns with both positive and negative offsets, indicating underlying periodic forces affecting the data, akin to an oscillator. The alignment between the lag feature and the actual values underscores a historical reliance or resilience of the present data points on past values. reflecting consistency in patterns.
DirectLLM's general annotations The provided time series data consists of 138 values and exhibits significant fluctuations. There are notable peaks at several points such as 118.0 and 113.0, indicating occasional high spikes in values. Conversely, values also drop to lows around 35.0, suggesting substantial variability. Most data points seem to oscillate around a mid-range, between approximately 50.0 and 70.0. This pattern suggests periods of stability interspersed with sporadic increases and decreases. Analyzing such a dataset can be indicative of a highly dynamic system or process with frequent transitions between states of varying magnitude.

Table 31: One more example of general annotation generation in the synthetic dataset. The selected dataset is shown in Fig. 9 (a). Purple highlights basic patterns, green highlights general patterns, and red highlights hallucinations. DirectLLM only captures on the basic trend pattern for time series annotation, while TESSA considers more important general patterns.

TESSA's general annotations The time series exhibits a complex behavior with an overall downward trend, accompanied by periods of fluctuation and temporary upward corrections. Key points include support and resistance levels, where the trend either pauses or reverses, notably around values 10.834 and 8.929. Increased volatility is evident, indicating active trading periods, which align with higher variability. Trend channels are apparent, where the data moves within upper and lower boundaries, particularly showing both descending and emerging upward trends. In some instances, breakout thresholds highlight significant changes, signaling momentum shifts.
DirectLLM's general annotations This uni-variate time series data exhibits a pattern with multiple phases of rise and fall, indicative of periodic fluctuations over time. Initially, the series starts at a moderate level, gradually ascending to a peak around the values of 7.58 and 7.52 before experiencing a gradual decline. The data then showcases another rise peaking just above 9 and 10 marks, followed by a sharp and continuous decline, entering negative territory around the value of -6.270 and continuing downwards. Near the end of the series, escalating towards positive values and climaxing at 5.091. The overall structure suggests well-defined periodic or seasonal trends, potentially influenced by external or inherent factors.

Table 32: One more example of general annotation generation in the synthetic dataset. The selected dataset is shown in Fig. 9 (b). Purple highlights basic patterns, green highlights general patterns, and red highlights hallucinations. DirectLLM only captures on the basic trend pattern for time series annotation, while TESSA considers more important general patterns.

TESSA's general annotations The time series data illustrates an overall upward trend characterized by increasing values over time, signifying growth. Initially, the series displays stability with minor fluctuations, often not dropping below certain support levels, indicating consistency. As the series advances, breakthrough points become apparent, where values exceed previous resistance levels, suggesting heightened momentum and possible shifts in liquidity that drive this progress. The later parts of the data feature more pronounced scending trends, pointing to a stronger upward movement. Periodic trend reversals also appear, reflecting temporary changes before returning to the dominant upward trend, which highlights the series' dynamic nature and potential for fluctuations.
DirectLLM's general annotations The given uni-variate time series data exhibits a fluctuating pattern with a general upward trend. Initially, from the first data point, there is a notable increase in values, reaching a peak around the seventh value. This is followed by a gradual decline and subsequent stabilization with minor fluctuations between the 12th and 31st values. Notably, around the 84th value, a significant surge in values begins, culminating in a prominent steep increase towards the end of the series, suggesting a potential exponential growth or shift occurring in the data. Overall, the time series transitions from more stable periods into a pronounced upward trend, signaling potential external influences or underlying factors driving the increase.

Table 33: One more example of domain-specific annotation generation in the Environment dataset. The selected dataset is shown in Fig. 10 (a). Purple highlights basic patterns, green highlights general patterns, and yellow highlights the domain-specific patterns.

TESSA's domain-specific annotations The air quality index (AQI) data exhibits significant fluctuations, with values ranging from 53 to 235 over time, indicating variability in air quality. Support levels around values like 100 and 140 suggest periods when air quality temporarily stabilizes or improves. On the other hand, resistance levels near values like 200 and 235 show points where air quality struggles to improve further before worsening. Several distinct upward trends, particularly from AQI values 70 to 150 and 177 to 235, indicate temporary periods of improvement in air quality, whereas downward trends around values 166 to 123 and 208 to 84 reflect deteriorating air quality after peaks. Monitoring these trends and critical thresholds will be essential for identifying and responding to significant pollution events effectively.
DirectLLM's domain-specific annotations The time series data represents the air quality index (AQI) over a series of observations, showing fluctuations in air pollution levels. Initially, the AQI values are moderate, transitioning to higher levels, peaking at alarming numbers such as 235 and 209, which indicate very unhealthy air quality. This indicates potential spikes in pollution that could be associated with environmental events or increased urban activity. Periods of lower AQI values suggest moments of improved air quality, but these are often followed by sharp increases, highlighting the inconsistency and poor air conditions in the observed timeframe. Overall, the data reflects significant air quality concerns, emphasizing the need for monitoring and potential interventions to safeguard public health.

Table 34: One more example of domain-specific annotation generation in the Environment dataset. The selected dataset is shown in Fig. 10 (b). Purple highlights basic patterns, green highlights general patterns, and yellow highlights the domain-specific patterns.

TESSA's domain-specific annotations The air quality index (AQI) exhibits noticeable upward and downward trends over time, indicating periods of improvement and decline in air quality. Regular seasonal patterns are apparent, with AQI values cyclically rising and falling, suggesting that certain times of the year may be more prone to higher pollution levels. Critical thresholds in AQI levels highlight points where air quality significantly improves or deteriorates, with some values acting as resistance levels that AQI struggles to surpass. Episodes of high variability in AQI indicate times of significant fluctuations, potentially due to varying pollution sources or climate conditions. Lastly, breakout patterns where AQI values suddenly shift indicate potential changes in pollution control effectiveness or new influences on air quality dynamics.
DirectLLM's domain-specific annotations The provided time series data represents fluctuations in the air quality index (AQI), with values indicating varying levels of air pollution over time. Initially, the AQI remains in a moderate range, typically below 100, but there is a noticeable spike as values reach up to 220, indicating very poor air quality. This suggests that certain periods experienced significantly higher pollution levels, which can have serious implications for public health and environmental quality. The data shows some improvements, but the peaks raise concerns about excessive pollution, particularly during certain months. Continuous monitoring and mitigation of pollution sources are essential to reduce the frequency and severity of these dangerous air quality levels.

Table 35: One more example of domain-specific annotation generation in the Energy dataset. The selected dataset is shown in Fig. 11 (a). Purple highlights basic patterns, green highlights general patterns, and yellow highlights the domain-specific patterns.

TESSA's domain-specific annotations Gas prices have exhibited an upward trend, peaking at $1.077 before experiencing a slight decline towards the latter part of the series. This indicates an initial growth phase followed by stabilization. Minor, yet consistent seasonal variations suggest periodic influences that slightly affect the price. The analysis of residuals reveals small but steady fluctuations, indicating some level of unpredictability in the market. Support and resistance analysis identifies critical thresholds around $1.003 and $1.092, highlighting potential reversal points and consolidation zones.
DirectLLM's domain-specific annotations The time series data represents fluctuations in gas prices over a period of time, listed in ascending order. Initially, the gas prices start around $1.06 but gradually show a downward trend, reaching as low as approximately $0.99 before slightly rising again. This decline suggests that there may have been favorable market conditions or increased supply during this period, which typically leads to lower prices for consumers. Toward the end of the observed period, prices stabilize around the $1.00 mark, indicating a potential plateau or gradual recovery in price levels. Overall, the data illustrates a dynamic market scenario, highlighting the volatility and changing nature of gas prices influenced by various factors in the energy domain.

Table 36: One more example of domain-specific annotation generation in the Energy dataset. The selected dataset is shown in Fig. 11 (b). Purple highlights basic patterns, green highlights general patterns, and yellow highlights the domain-specific patterns.

TESSA's domain-specific annotations
Gas prices show a generally upward trend, starting around 1.11 and gradually increasing over time, indicating a long-term positive movement. Minor seasonal fluctuations are observed, but they do not dominate the overall trend. There are key support levels around 1.113 and resistance levels at 1.133, which may act as pivotal points for future price movements. Residuals suggest occasional minor deviations from the trend due to random factors, resulting in a generally stable series with slight intraday volatility. The moving average values reinforce this steady climb, suggesting continued stability with periodic minor disruptions in gas prices.
DirectLLM's domain-specific annotations
The time series data represents fluctuations in gas prices over a specific period, with values consistently hovering around the $1.11 to $1.18 range. Notably, the data shows slight increases and decreases in price, suggesting moderate volatility within this timeframe. The highest recorded price peaks at approximately $1.18, reflecting potential market adjustments or external factors influencing gas prices. This stability could indicate a balanced supply and demand scenario, although one should remain aware that various events, such as geopolitical developments, can lead to sudden shifts. Overall, these insights provide a snapshot of gas price trends, useful for consumers and industry stakeholders in making informed decisions related to energy expenditures.

Table 37: One more example of domain-specific annotation generation in the Social Good dataset. The selected dataset is shown in Fig. 12 (a). Purple highlights basic patterns, green highlights general patterns, and yellow highlights the domain-specific patterns.

TESSA's domain-specific annotations
The unemployment rate data shows a gentle upward trend from approximately 5.4% to 6.57% over the observed period, indicating a consistent rise in unemployment levels. This gradual increase implies a weakening labor market, which could be influenced by broader socioeconomic conditions. Although there are oscillations suggesting cyclical patterns, these are not strong and exhibit some irregularities, pointing to potential short-term fluctuations or external disruptions. The stable relationship between consecutive data points suggests that the unemployment rate changes are relatively steady without abrupt shifts, allowing for some degree of predictability. Overall, the trend may signal policymakers to consider implementing measures to address the rising unemployment, especially if the current trajectory continues.
DirectLLM's domain-specific annotations The time series data represents the unemployment rate over a period of time. Initially, the unemployment rate showed a slight fluctuation around the 5% mark, indicating a relatively stable job market. However, a noticeable increase began, with rates rising to over 7%, suggesting a worsening economic situation and potential job losses. After peaking at 7.3%, the data indicates a gradual decrease, though the rates remained relatively high around 6.5% towards the end. This pattern highlights a period of economic challenge, with gradual steps towards recovery.

Table 38: One more example of domain-specific annotation generation in the Social Good dataset. The selected dataset is shown in Fig. 12 (b). Purple highlights basic patterns, green highlights general patterns, and yellow highlights the domain-specific patterns.

TESSA's domain-specific annotations The unemployment rate data reflects a positive economic momentum, as evidenced by a persistent upward trend starting from around 4.1 and moving above 6, suggesting a gradual increase in unemployment over the series. This trend could indicate rising employment participation pressure or an effectiveness of policy breakthrough effects aimed at fostering job growth, which could be interpreted as a counter-effect if not analyzed in tandem with job creation data. Despite the upward trend, the series shows seasonality patterns, with periodic fluctuations above and below a baseline, suggesting regular influences on employment rates, possibly due to economic cycles or policy adjustments. These fluctuations may align with safety net thresholds, which temporarily reduce unemployment rates during specific intervals. In summary, the data represents a complex interplay of economic dynamics, signaling opportunities for policy interventions to maintain a balance between growth and employment stability.

DirectLLM's domain-specific annotations

The time series data illustrates the fluctuation of the unemployment rate over a specific period. Initially, the unemployment rate started at a level of 4.5% and experienced a gradual decline to a low of about 3.6%. However, this was followed by an upward trend, eventually reaching highs of around 6.3%. Such fluctuations might have been influenced by changing economic conditions, labor market policies, or external events impacting employment. Notably, the trend indicates periods of economic strengthening followed by downturns, reflecting possible cycles of growth and contraction in the job market.

```
System Prompt

Your job is to act an extremely clever time series expert to remove
the domain-specific terms from the given textual descriptions. You
will write a high-quality domain-decontextulized time series
annotation that is informative and helps in understanding given time
series data.
```

```
User Prompt

Your task is to remove the domain-specific [example of domain-
specific terms] description from the given texts. Please use some
time series analysis common knowledge to replace the domain-specific
term with terms common-used in time series analysis. Write a concise
report that provides insights crucial for understanding the time
series data. The textual annotation is:
                    [Textual annotation]
Please do not include some domain-specific information and terms in
[specific domain] to write the report. The indicators name should
also not be included in the report. The text data is only used to
understand the time series and generate more insightful and concise
analysis.
```

Figure 13: Prompt for domain decontextualization.

Table 39: Prompt for text feature extraction

System Prompt Your job is to an exceptionally clever time series expert to extract time-series features from the textual annotations.

User Prompt: Your task is to extract the text-wise features based on the given textual annotation about the time series annotation. Each text annotation is separated by a '|' token:

[Decontextualized textual annotation]

Based on the textual annotation, please use common knowledge of time series analysis to extract the text-wise features that are explictly or implictly mentioned in the textual annotations but missing in the following time-series-wise tokens (separated by '|'):

[Text-wise features]

The extracted features should be concise and common-used feature terms for time series analysis. Please only output the extracted features in the format of python list.

System Prompt

Your job is to an exceptionally clever time series expert to score the given time series features by the relevance between the time-series features and the textual annotations

User Prompt

Your task is to score and rank the given time series features based on the given textual annotations. The candidate time series features are separated by a '|' tokes:
[Time-series-wise features]
Each text annotation is seperated are by a '|' tokes:
[Decontextualized textual annotations]
Based on the textual annotation, please use common knowledge of time series analysis to provide a score and rank of given time-series feature names, based on the following score metric: (1) Every time the time-series feature is explicitly appear in one annotation, the score of this time-series feature add 1. (2) Every time the time-series feature is implicitly appear in one annotation, that is, although the time-series feature is not explicitly mentioned in annotation, the internal properties are implied in annotation, then the score of this time-series feature add 0.5. Note that please only output the scores of the features that in the candidate time series features.

Figure 14: Prompt for scoring time-series-wise feature importance

System Prompt

Your job is to an exceptionally clever time series expert to score the given text-wise features by the relevance between the text-wise features and time series-wise features

User Prompt

Your task is to score and rank the given text-wise features based on the given time series-wise features. The candidate text-wise features are separated by a '|' tokes :
[Text-wise features]
Based on the textual annotation, please use common knowledge of time series analysis to extract the time series features that are explicitly or implicitly mentioned in the textual annotations but missing in the following time-series-wise tokens (separated by '|'):
[Time-series-wise features]
Based on the time series-wise features, please use common knowledge of time series analysis to provide a score and rank of given time-series feature names, based on the following score metric: (1) For each candidate text-wise feature, if it has already explicitly/implicitly appeared in the given time series-wise features, the score of this text-wise features are fixed as -2. (2) For each candidate text-wise feature that satisfies (1), that is, doesn't have overlap with time series-wise features, if the text-wise feature is related to time series analysis, then the score of this time-series feature add 1. Note that please only output the scores of the features that in the candidate time series features.

Figure 15: Prompt for scoring text-wise feature importance

System Prompt

Your job is to an exceptionally clever time series expert to do time series annotation by explaining the given time series data. You will write a high-quality annotation that is informative and helps in understanding given time series data

User Prompt

You will be provided a univariate time series data, several time series features extracted from the given univariate time series data and corresponding text-wise token related to time series. Your task is to interpret the given univariate time series data based on the given time series tokens and features. The generated interpretation is for general audiences without any time series knowledge. Assume we don't know domain and background knowledge. For the original univariate time series, each value in these time series features is separated by a '|' token:

[Time series data]

For the intra-variable time series-wise features in all variables, each value in these time series features is separated by a '|' token:

[Time-series-wise features]

We also have several text-wise tokens as references for time series interpretation, they are separated by a '|' token:

[Time-series-wise features]

Based on these time series features and text-wise tokens, write a concise interpretation that provides insights crucial for understanding the given time series data. These tokens are only help you to write a more insightful annotation to help users to understand the time series, you are not required to add them into the interpretation but just consider to analysis these tokens. Please avoid using uncommon words and phrases. Your interpretation should be up to five sentence, yet comprehensive.

Figure 16: Prompt for general annotation

System Prompt

Your job is to an an extremely clever clever time series expert to extract domain-specific terms related to the given time series features.

User Prompt

Your task is to extract domain-specific terms related to the given time series features for the [target domain name]. domain, which is useful for time series analysis based on the given time series features. The features related to time series analysis are separated by a '|' tokens:

[Extracted general features]

Each given textual annotation is separated by a '|' token:

[Target-domain annotations]

The time series data is used to denote [target variable name]. Your extracted domain-specific terms should be relevant and useful to the given time series tokens, which are useful in understanding the time series data and doing time series analysis. These extracted domain-specific terms should be the time series feature for time series analysis. Please use domain-specific descriptions to describe the extracted term instead of directly using general time series feature tokens.

Figure 17: Prompt for jargon extraction

System Prompt

Your job is to an extremely clever expert to write interpretations of the time series data for [target domain name] domain.

User Prompt

You will be provided a comprehensive general time series interpretation that contains rich information. Several human-written domain-specific interpretations in the [target domain name] domain are also given as the reference annotations. Your task is to write domain-specific interpretations for [target variable name] in [target domain name] domain for general audiences, all necessary important variables and patterns are provided and extracted from the given general annotation.

Your writing style need to refer to the given reference domain-specific annotation:

[General annotation]

The reference domain-specific interpretation is:

[Reference annotation]

For the given general interpretation:

[Definition of each variable].

The key variable is [Key variable name] in focus. The domain-specific terms are separated by a '|' token:

[Domain-specific terms].

You are required to select several the most important patterns, variables, text-wise tokens analysis and inter-variable relationships from the general interpretations to write the domain-specific interpretations. Note that please ignore the less important variables, no-correlated patterns and the description no related to the analysis in the generated annotations. Also only include strong correlation patterns and important variables in the annotations. The domain-specific terms are only help you to write a more informative interpretations to help users to understand the time series, you are not required to involved them. The reference interpretations are only help you to refer the writing style and the involved information. The written interpretation should be naturally and smoothly, which sounds like written by domain experts. Please avoid using uncommon words and phrases. Your report should be limited to five sentences, yet comprehensive.

Figure 18: Prompt for domain-specific annotation

System Prompt

Your job is to an exceptionally clever time series expert with the unique ability to find errors and repeat features from the given domain-specific time series annotations and provide feedback to help revise them.

User Prompt

You will be provided with a general annotation, a corresponding domain-specific annotation in [target domain name] domain and extracted domain-specific terms. Your task is to use common knowledge of time series analysis to review the annotation, identify if the domain-specific annotation is contextual consistent with the general annotation and the domain-specific terms are well involved in the domain-specific annotation.

The domain-specific annotation is:

[domain-specific annotation]

The general annotation is:

[general annotation]

Please provide comments about improving the process of extracting domain-specific terms and converting general annotation to domain-specific annotation such that LLM can perform better based on the comments.

Figure 19: Prompt for reviewing annotation

Table 40: Prompt for evaluating clarity of the annotations of time series

System Prompt Your job is to act as an extremely clever time series expert to scoring the clarity and readability of textual interpretations generated for time series data. You will assess how clear, concise, and understandable the interpretation is. You will score each interpretation on a scale of 1 to 5, where 1 indicates that the interpretation is unclear and difficult to understand, and 5 indicates that the interpretation is exceptionally clear and easy to read. Your evaluation should consider factors such as language clarity, coherence, and how effectively the interpretation communicates the insights from the time series data.
User Prompt: You will be provided two time series interpretations. Your task is to evaluate and compare the comprehensiveness of the following interpretation of time series data. The time series interpretation A is shown as follow: [input annotation A]. The time series interpretation B is shown as follow: [input annotation B]. Assign a score between 1 and 5 for each interpretation based on the following criteria: 1. Score 1: The interpretation is poorly written, confusing, or unclear. It is difficult to follow the logic, and essential insights such as trends, seasonality, or patterns are not communicated effectively due to poor readability. 2. Score 2: The interpretation is somewhat understandable but still suffers from clarity issues. The language may be overly technical, vague, or lacking coherence, making it hard to extract insights from the explanation. 3. Score 3: The interpretation is moderately clear but could be improved in readability. While it conveys some important insights (e.g., trends, seasonality, etc.), the structure may be inconsistent, or the explanation may use technical jargon that hinders understanding. 4. Score 4: The interpretation is clear, readable, and communicates most of the insights effectively. 5. Score 5: The interpretation is clear and readable,. Advanced terminology is used appropriately to enhance the clarity of the insights, and the interpretation draws meaningful connections between observed data patterns and broader market or system behaviors (e.g., support-level, resistant-level, trend reversals)..

Table 41: Prompt for evaluating comprehensiveness of annotations of synthetic time series data

System Prompt Your job is to act as an extremely clever time series expert to scoring the comprehensiveness of two textual interpretations generated for time series data. Your primary focus is on assessing how well each interpretation covers important patterns within the data. These patterns may include, but are not limited to, seasonality, trend, residue, frequency, lag features, and rolling window features. Additionally, for multivariate time series data, you should evaluate whether the interpretation identifies inter-variable patterns, such as correlations. The more patterns an interpretation covers, the higher the score it should receive. You will score each interpretation separately on a scale of 1 or 5.
User Prompt: You will be provided a time series interpretation. Your task is to evaluate and compare the clarity and readability of the following interpretation of time series data. The time series interpretation A is shown as follow: [input annotation A]. The time series interpretation B is shown as follow: [input annotation B].The ground-truth annotation is shown as follows: [ground truth annotations]. Consider the following patterns when evaluating each interpretation: 1. Seasonality. 2. Trend; 3. Residue; 4. Fourier Feature; 5. Lag features; 6. Rolling window features; Assign a score 1 or 5 for each interpretation based on the following criteria: if interpretation A implicitly or explicitly cover more above patterns (e.g., 1. Seasonality. 2. Trend; 3. Residue; 4. Fourier Feature; 5. Lag features; 6. Rolling window features;) than B, score A to 5 and B to 1. Otherwise, score A to 1 and B to 5.

Table 42: Prompt for evaluating clarity of the annotations of time series

System Prompt Your job is to act as an extremely clever time series expert to score the clarity and readability of textual interpretations generated for time series data. You will assesse how clear, concise, and understandable the interpretation is. You will score each interpretation on a scale of 1 to 5, where 1 indicates that the interpretation is unclear and difficult to understand, and 5 indicates that the interpretation is exceptionally clear and easy to read. Your evaluation should consider factors such as language clarity, coherence, and how effectively the interpretation communicates the insights from the time series data.
User Prompt: You will be provided two time series interpretations. Your task is to evaluate and compare the comprehensiveness of the following interpretation of time series data. The time series interpretation A is shown as follows: [input annotation A]. The time series interpretation B is shown as follows: [input annotation B]. Assign a score between 1 and 5 for each interpretation based on the following criteria: 1. Score 1: The interpretation covers very few or none of the important patterns in the time series data. It fails to address key aspects such as seasonality, trends, residues, frequency, or correlations in multivariate data. 2. Score 2: The interpretation covers a few important patterns, but significant aspects are missing or poorly addressed. It provides a limited view of the time series data. 3. Score 3: The interpretation covers some important patterns but misses or inadequately addresses others. It gives a moderate level of insight into the time series data but lacks full comprehensiveness. 4. Score 4: The interpretation covers most of the important patterns, including seasonality, trends, residues, frequency, and correlations in multivariate data. It is comprehensive but may have minor omissions or weaknesses. 5. Score 5: The interpretation is highly comprehensive, covering all important patterns in the time series data. It thoroughly addresses seasonality, trends, residues, frequency, and correlations in multivariate data without significant omissions.

Table 43: Prompt for evaluating comprehensiveness of domains-specific annotations

System Prompt Your job is to act as an extremely clever time series expert to scoring the comprehensiveness of two textual interpretations generated for time series data. Your primary focus is on assessing how well each interpretation covers important patterns within the data. These patterns may include, but are not limited to, seasonality, trend, residue, frequency, lag features, and rolling window features. Additionally, for multivariate time series data, you should evaluate whether the interpretation identifies inter-variable patterns, such as correlations. The more patterns an interpretation covers, the higher the score it should receive. You will score each interpretation separately on a scale of 1 to 5, where 1 indicates minimal pattern coverage, and 5 indicates highly comprehensive coverage of the data's important patterns. You will score each interpretation separately on a scale of 1 to 5.
User Prompt: You will be provided a time series interpretation. Your task is to evaluate and compare the clarity and readability of the following interpretation of time series data. The time series interpretation A is shown as follow: [input annotation A]. The time series interpretation B is shown as follow: [input annotation B]. Consider the following patterns when evaluating each interpretation: 1. Seasonality. 2. Trend; 3. Residue; 4. Frequency; 5. Lag features; 6. Rolling window features; 7. For multivariate data: Inter-variable correlations. Assign a score between 1 and 5 for each interpretation based on the following criteria: 1. Score 1: The interpretation covers few or none of the important patterns. It is largely incomplete. 2. Score 2: The interpretation covers some patterns but misses many others, providing only a basic overview. 3. Score 3: The interpretation covers several important patterns but is still incomplete, missing key aspects of the data. 4. Score 4: The interpretation covers most of the important patterns, with only minor omissions. It provides a strong overview of the data. 5. Score 5: The interpretation comprehensively covers all important patterns, including any inter-variable correlations in multivariate data. It provides a thorough and complete analysis.

Table 44: Prompt for evaluating domain-relevance of domains-specific annotations

System Prompt Your job is to act as an extremely clever time series expert to scoring the domain relevance of two textual interpretations generated for time series data. Your primary focus is on assessing how well each interpretation aligns with the established principles and practices specific to time series analysis. This includes evaluating the correctness of the methods used, the appropriateness of the terminology, and the accuracy of the interpretation of patterns within the time series data. You will score each interpretation separately on a scale of 1 to 5, where 1 indicates poor alignment with time series analysis principles, and 5 indicates a highly accurate and relevant interpretation that effectively applies time series analysis concepts
User Prompt: You will be provided two time series interpretation for one time series data. Your task is to evaluate and compare the domain relevance of the following two interpretation of time series data, focusing on their alignment with established principles and practices in time series analysis. The time series interpretation A is shown as follow: [input annotation A]. The time series interpretation B is shown as follow: [input annotation B]. Consider the following patterns when evaluating each interpretation: 1. Correct use of time series analysis terminology (e.g., seasonality, trend, autocorrelation); 2. Accurate application of time series analysis methods; 3. Appropriate interpretation of patterns within the time series data; 4. Relevance to the time series context and best practices; Assign a score between 1 and 5 for each interpretation based on the following criteria: 1. Score 1: The interpretation uses incorrect or inappropriate time series analysis terminology and methods. It misinterprets the data and lacks relevance to established practices. 2. Score 2: The interpretation uses some correct terminology and methods but is often inaccurate or lacks contextual relevance to time series analysis. It partially aligns with the principles but has significant gaps. 3. Score 3: The interpretation correctly uses several time series analysis terms and methods but lacks full accuracy or completeness. It is moderately relevant but still has gaps in applying time series principles. 4. Score 4: The interpretation accurately applies most time series analysis terminology and methods, with only minor errors. It is contextually relevant and appropriate, aligning well with best practices. 5. Score 5: The interpretation is highly accurate in its use of time series analysis terminology and methods, with a strong contextual relevance. It demonstrates a deep understanding of time series analysis and applies concepts correctly and comprehensively.

Table 45: Prompt for evaluating clarity of annotations generated by TESSA and TESSA/F.

System Prompt Your job is to act as an extremely clever time series expert to scoring the clarity and readability of textual interpretations generated for time series data. You will assess how clear, concise, and understandable the interpretation is. You will score each interpretation on a scale of 1 to 5, where 1 indicates that the interpretation is unclear and difficult to understand, and 5 indicates that the interpretation is exceptionally clear and easy to read. Your evaluation should consider factors such as language clarity, coherence, and how effectively the interpretation communicates the insights from the time series data.
User Prompt: You will be provided two time series interpretations. Your task is to evaluate and compare the comprehensiveness of the following interpretation of time series data. The time series interpretation A is shown as follow: [input annotation A]. The time series interpretation B is shown as follow: [input annotation B]. Assign a score between 1 and 5 for each interpretation based on the following criteria: 1. Score 1: The interpretation covers very few or none of the important patterns in the time series data. It fails to address key aspects such as seasonality, trends, residues, frequency, or correlations in multivariate data. 2. Score 2: The interpretation covers a few important patterns, but significant aspects are missing or poorly addressed. It provides a limited view of the time series data. 3. Score 3: The interpretation covers some important patterns but misses or inadequately addresses others. It gives a moderate level of insight into the time series data but lacks full comprehensiveness. 4. Score 4: The interpretation covers most of the important patterns, including seasonality, trends, residues, frequency, and correlations in multivariate data. It is comprehensive but may have minor omissions or weaknesses. 5. Score 5: The interpretation is highly comprehensive, covering all important patterns in the time series data. It thoroughly addresses seasonality, trends, residues, frequency, and correlations in multivariate data without significant omissions.