Empowering Time Series Analysis with Large Language Models: A Survey

Yushan Jiang[1], Zijie Pan[1], Xikun Zhang[2], Sahil Garg[3], Anderson Schneider[3], Yuriy Nevmyvaka[3], Dongjin Song[1]
[1]School of Computing, University of Connecticut, USA
[2]School of Computer Science, The University of Sydney, Australia
[3]Department of Machine Learning Research, Morgan Stanley, USA
{yushan.jiang, zijie.pan, dongjin.song}@uconn.edu, xzha0505@uni.sydney.edu.au,
{sahil.garg, anderson.schneider, yuriy.nevmyvaka}@morganstanley.com,

Abstract

Recently, remarkable progress has been made over large language models (LLMs), demonstrating their unprecedented capability in varieties of natural language tasks. However, completely training a large general-purpose model from the scratch is challenging for time series analysis, due to the large volumes and varieties of time series data, as well as the non-stationarity that leads to concept drift impeding continuous model adaptation and retraining. Recent advances have shown that pre-trained LLMs can be exploited to capture complex dependencies in time series data and facilitate various applications. In this survey, we provide a systematic overview of existing methods that leverage LLMs for time series analysis. Specifically, we first state the challenges and motivations of applying language models in the context of time series as well as brief preliminaries of LLMs. Next, we summarize the general pipeline for LLM-based time series analysis, categorize existing methods into different groups (*i.e.*, direct query, tokenization, prompt design, fine-tune, and model integration), and highlight the key ideas within each group. We also discuss the applications of LLMs for both general and spatial-temporal time series data, tailored to specific domains. Finally, we thoroughly discuss future research opportunities to empower time series analysis with LLMs.

1 Introduction

In the past few years, significant advances have been made in large language models (LLMs), taking artificial intelligence and natural language processing a giant leap forward. LLMs, *e.g.*, OpenAI's GPT-3 and Meta's Llama 2 [Touvron *et al.*, 2023b], have not only exhibited an unparalleled ability to create narratives that are both coherent and contextually relevant but also demonstrated their remarkable accuracy and proficiency in complex and nuanced tasks such as responding to queries, translating sentences between multiple languages, code generation, and so on.

Inspired by the success of LLMs, a great deal of effort has been made to train general-purpose time series analysis models [Wu *et al.*, 2022; Garza and Mergenthaler-Canseco, 2023] to facilitate various underlying tasks, such as classification, forecasting, and anomaly detection. These efforts, however, are hindered by two key challenges. First, time series data may come in various forms - univariate or multivariate, for example - in large volumes, and from a variety of domains: healthcare, finance, traffic, environmental sciences, *etc*. This increases the complexity of model training and makes it difficult to handle different scenarios. Second, real-world time series data often exhibit non-stationary properties when they are continuously accumulated/collected, meaning that the statistical characteristics of time series data, such as mean, variance, and auto-correlation, will change over time. This phenomenon is common in applications such as financial markets, climate data, and user behavior analytics where patterns and trajectories evolve and do not remain constant. It can lead to the concept drift problem, as the statistical properties of the target variables may also change over time, making it difficult for the large models to be continuously adapted and re-trained [Kim *et al.*, 2021].

More recently, instead of training a general-purpose time series analysis model from the scratch, there is an increasing

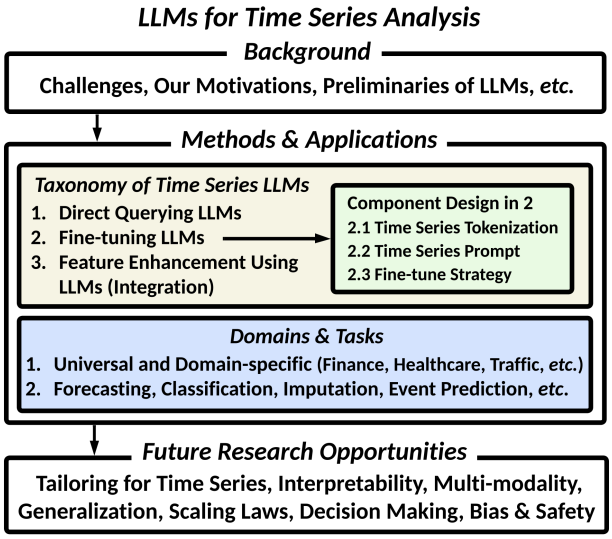

Figure 1: The framework of our survey

trend in exploiting existing LLMs in various time series applications. Consequently, different methodologies have been developed based on application types. In this survey, we provide a comprehensive and systematic overview of existing methods that leverage LLMs for time series analysis. As shown in Figure 1, we will first discuss the challenges, motivations, and preliminaries of LLMs. Next, we will summarize the general pipeline for LLM-based time series analysis and introduce five different types of techniques for applying LLMs: direct query, tokenization, prompt design, fine-tuning, and model integration. We will also discuss application of LLMs to specific domains. For better comparison, we provide a comprehensive table that summarizes representative methods, their modeling strategies, associated tasks and domains (as shown in Table 1). Finally, we highlight potential future research opportunities to further advance time series analysis with LLMs. We also provide the up-to-date resources in the GitHub repository[1]. In summary, the main contributions of this survey include:

- We catalog papers on LLM-based time series analysis that cover representative approaches since 2022.
- We systematically survey existing methods that leverage LLMs for time series analysis, uniquely categorize them into five groups based on the methodology, and discuss their application tasks and domains.
- We discuss and highlight future directions that advance time series analysis with LLMs and encourage researchers and practitioner to further investigate this field.

2 Background

2.1 General Large Language Models

Early advancements in natural language processing involve neural language models (NLMs) [Arisoy *et al.*, 2012] and pioneering LLMs, such as GPT-2 [Radford *et al.*, 2019], BERT [Devlin *et al.*, 2018], RoBERTa [Liu *et al.*, 2019], and XLNet [Yang *et al.*, 2019]. More recently, the rise of more powerful LLMs (*e.g.,* multi-modal large language models [Yin *et al.*, 2023]) has revolutionized AI usage because of their exceptional ability to handle complex tasks. We adopt a similar criterion to that in [Zhao *et al.*, 2023; Jin *et al.*, 2023] and divide LLMs into two categories: embedding-visible LLMs and embedding-invisible LLMs. The embedding-visible LLMs are usually open-sourced with inner states accessible. Notable examples include T5 [Raffel *et al.*, 2020], Flan-T5 [Chung *et al.*, 2022], LLaMA [Touvron *et al.*, 2023a; Touvron *et al.*, 2023b], ChatGLM [Du *et al.*, 2022], *etc*. These open-sourced LLMs are adaptable for various downstream tasks, demonstrating impressive capabilities in both few-shot and zero-shot learning settings, without the need to retrain from scratch. On the other hand, the embedding-invisible LLMs are typically closed-sourced with inner states inaccessible to the public. This type of LLMs include PaLM [Chowdhery *et al.*, 2023], GPT-3 [Brown *et al.*, 2020], GPT-4 [Achiam and et al., 2023]. For these models, researchers are limited to conducting inference tasks through prompting via the API calls. These LLMs can be potentially exploited for time series analysis.

[1]https://github.com/UConn-DSIS/Empowering-Time-Series-Analysis-with-LLM

2.2 Leveraging LLMs in Time Series Analysis

The rapid development of LLMs in natural language processing has unveiled unprecedented capabilities in sequential modeling and pattern recognition. It is natural to ask: How can LLMs be effectively leveraged to advance general-purpose time series analysis?

Our survey aims to answer the question based on a thorough overview of existing literature. We claim that LLMs can serve as a flexible as well as highly competent component in the time series modeling. The flexibility lies in a wide spectrum of available LLMs that can be employed and the variety of ways they can be configured for time series analysis (Section 3). Regarding their competence, LLMs can be tailored for a wide range of real-world applications with domain-specific context (Section 4). Certainly, there still exists several challenges in this field and we discuss future opportunities (Section 5).

Next, we highlight the difference between our survey and a few recent relevant ones in terms of the scope and focus. Deldari [2022] and Ma *et al.* [2023] both include the summary of pre-trained techniques for time series where Deldari [2022] is specialized in self-supervised representation learning (SSRL) methods for multi-modal temporal data (not only time series). Mai *et al.* [2023b] summarizes large pre-trained models (including LLMs) for time series in geospatial domain. Jin *et al.* [2023] provides a comprehensive survey of large pre-trained models for time series and general spatial-temporal data. Compared with [Jin *et al.*, 2023; Mai *et al.*, 2023b], our survey focuses on **LLMs** for **time series analysis**, which is the only one categorizing existing methods based on modeling strategy. Our survey is also uniquely positioned to provide detailed introductions of not only universal methodology design but also various applications with domain-specific context. Figure 2 and Table 1 demonstrate our uniqueness.

3 Taxonomy of LLMs in Time Series Analysis

In this section, we conduct a detailed discussion of existing research that utilizes LLMs for universal time series modeling and thoroughly analyze the design of their components, where we categorize and brief the designs of domain-specific methods. We will also elaborate by tailoring them to specific domain contexts in Section 4. The detailed taxonomy is provided in Table 1.

General Pipeline of LLMs. To adopt LLMs for time series analysis, three primary methods are employed: direct querying of LLMs (Section 3.1), fine-tuning LLMs with tailored designs (Section 3.2-3.4), and incorporating LLMs into time series models as a means of feature enhancement (Section 3.5). Specifically, three key components can be leveraged to fine-tune LLMs as shown in Figure 2: The input time series are first tokenized into embedding based on proper tokenization techniques, where proper prompts can be adopted2 to further enhance the time series representation. As such,

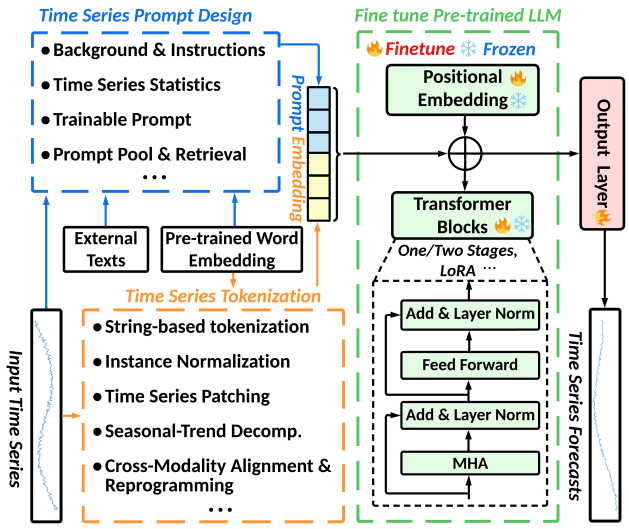

Figure 2: Categorization of component design for fine-tuning time series LLMs (Section 3.2-3.4)

LLMs can better comprehend prompt-enhanced time series embedding and be fine-tuned for downstream tasks, based on sophisticated strategies.

3.1 Direct Query of LLMs

PromptCast [Xue and Salim, 2023] is the first work that directly conducts general time series forecasting in sentence-to-sentence fashion using pre-trained LLMs. It introduces a novel forecasting setting, *i.e.*, prompt-based time series forecasting that embeds lag information as well as instructions into the prompts and uses output sentences from the LLMs to conduct forecasting. Directly querying LLMs can also be beneficial in domain-specific scenarios [Yu *et al.*, 2023; Wang *et al.*, 2023], particularly when leveraging advanced pre-trained LLMs (*e.g.,* GPT-4 [Achiam and et al., 2023] and OpenLLaMA [Geng and Liu, 2023; Computer, 2023; Touvron *et al.*, 2023b]) in conjunction with context-inclusive prompts that provide relevant domain knowledge.

While direct usage of LLMs for time series forecasting can be zero-shot or few-shot, instruction-based fine-tuning, and Chain-of-Thoughts (COT) [Lightman *et al.*, 2023; Wei *et al.*, 2022; Zhang *et al.*, 2023b] have shown positive effects on the reasoning process. LLMTime [Gruver *et al.*, 2023] also demonstrates that LLMs are effective zero-shot time series learners with proper text-wise tokenization on time series.

3.2 Time Series Tokenization Design

The aforementioned works [Xue and Salim, 2023; Yu *et al.*, 2023; Gruver *et al.*, 2023] convert numerical values of time series data into string-based tokens so that LLMs can seamlessly encode time series as the natural language inputs. In this subsequent section, we exclusively focus on the tokenization design to represent time series data more effectively. In practical applications, time series analysis often encounters the challenge of distribution shifts. To address this issue, major works adopt channel independence and reversible instance normalization (RevIN) [Kim *et al.*, 2021] before tokenization.

Patch representation [Nie *et al.*, 2023] for time series has shown promising results in time series analysis with transformer-based models. For a univariate time series input with length L: $\mathbf{X}_{1\mathrm{D}} \in \mathbb{R}^L$, the patching operation first repeats the final value in the original univariate time series S times. Then, it unfolds the input univariate time series through a sliding window with the length of patch size P and the stride size of S. Through patching, the univariate time series will be transformed into two-dimensional representations $\mathbf{X}_{\mathrm{p}} \in \mathbb{R}^{P \times N}$, where N is the number of patches with $N = \left\lfloor \frac{(L-P)}{S} \right\rfloor + 2$. It can be mathematically formulated as:

$$\mathbf{X}_{\mathrm{p}} = \mathrm{Unfold}\left(\mathrm{Right\_Pad}\left(\mathbf{X}_{1\mathrm{D}}\right), \mathrm{size} = P, \mathrm{stride} = S\right) \quad (1)$$

Patching tokenization design preserves the original relative order of the data and aggregates local information into each patch. One Fits All (OFA) [Zhou *et al.*, 2023], Time-LLM [Jin *et al.*, 2024] and other works [Chang *et al.*, 2023; Sun *et al.*, 2024; Cao *et al.*, 2024; Liu *et al.*, 2023b; Bian *et al.*, 2024; Jia *et al.*, 2024; Pan *et al.*, 2024] primarily adopt this method to tokenize time series data. Prior to tokenization, to enhance the characterization of time series data, an additive decomposition method can be used to extract trend, seasonal, and residual components from the original time series data, *i.e.*, $\mathbf{X}_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}}^T + \mathbf{X}_{\mathrm{t}}^S + \mathbf{X}_{\mathrm{t}}^R$. Either classical additive decomposition or additive STL decomposition [Cleveland *et al.*, 1990] can be used to extract corresponding components [Pan *et al.*, 2024; Cao *et al.*, 2024].

In order to harmonize the modalities of numerical data and natural language, an auxiliary loss has been introduced by TEST [Sun *et al.*, 2024] to enhance the cosine similarity between the embeddings of time series tokens and selected text prototypes, as well as to ensure proximity in the textual prototype space for similar time series instances. For a similar purpose, Time-LLM [Jin *et al.*, 2024] proposes to use a multi-headed attention mechanism to align the patched time series representation with the pre-trained text prototype embedding, acquired through linear probing. Specifically, Time-LLM reprograms time series patches in each attention head via:

$$\mathbf{Z}_k^{(i)} = \mathrm{Softmax}\left(\frac{\mathbf{Q}_k^{(i)}\mathbf{K}_k^{(i)\top}}{\sqrt{d_k}}\right)\mathbf{V}_k^{(i)} \quad (2)$$

where query matrices $\mathbf{Q}_k^{(i)} = \hat{\mathbf{X}}_p^{(i)}\mathbf{W}_k^Q$, key matrices $\mathbf{K}_k^{(i)} = \mathbf{E}'\mathbf{W}_k^K$, value matrices $\mathbf{V}_k^{(i)} = \mathbf{E}'\mathbf{W}_k^V$, and $\mathbf{E}'$ is the reduced-sized pre-trained word embedding.

In addition to patch-based tokenization approaches, Chronos [Ansari *et al.*, 2024] employs bin-based quantization to convert numerical values into discrete tokens, UniTime [Liu *et al.*, 2023b] fuses the time series input with random binary masks and adopts another reconstruction objective to enhance the representations.

3.3 Prompt Design

PromptCast [Xue and Salim, 2023] develops template-based prompts for LLM time series forecasting, while some methods [Yu *et al.*, 2023; Xue *et al.*, 2022a; Wang *et al.*, 2023; Liu *et al.*, 2023a; Jia *et al.*, 2024; Liu *et al.*, 2023b] enrich the prompt design by incorporating LLM-generated or

gathered background information, which highlights the importance of context-inclusive prompts in real-world applications. Besides the background and instruction prompts, Time-LLM [Jin *et al.*, 2024] adds statistical information of the time series data to facilitate time series forecasting. Compared with fixed and non-trainable prompts, a soft and trainable prompt makes it easier for LLMs to understand and align with the input [Lester *et al.*, 2021]. Prefix soft prompts are the task-specific embedding vectors, learned based on a loss from LLMs' output and the ground truth.

TEST [Sun *et al.*, 2024] initializes the soft prompts with uniform distributions, text embedding of the downstream task labels, or the most common words from the vocabulary. TEMPO [Cao *et al.*, 2024] and S^2IP-LLM [Pan *et al.*, 2024] focus on retrieval-based prompt design. The former selects highly representative soft prompts from key-value pools, while the latter chooses the most similar semantic anchors derived from pre-trained word embeddings for fine-tuning through a similarity score matching mechanism.

3.4 Fine-tuning Strategy

Fine-tuning pre-trained LLMs is pivotal to leveraging LLMs' strong pattern recognition and reasoning capabilities to facilitate downstream tasks. Several existing works opt to fine-tune pre-trained LLMs directly (one stage) for time series analysis. The main difference lies in how the modules' parameters are updated during the fine-tuning process. As a standard practice [Lu *et al.*, 2022; Houlsby *et al.*, 2019], OFA [Zhou *et al.*, 2023] and S^2IP-LLM [Pan *et al.*, 2024] fine-tune the positional embedding and layer normalization layers, and freezes self-attention layers and Feedforward Neural Networks (FFN) as they contain majorities of the learned knowledge. LLM4TS [Chang *et al.*, 2023] and TEMPO [Cao *et al.*, 2024] further fine-tune the self-attention modules using Low-Rank Adaptation (LoRA) [Hu *et al.*, 2021] by introducing trainable low-rank bypasses to the query (Q) and key (K) matrices in the self-attention mechanism. Instead of directly modifying the original weight matrices W^Q and W^K, LoRA introduces A^Q, A^K, B^Q, and B^K, which are much smaller in size compared to W^Q and W^K. The modified query and key matrices in LoRA can be represented as:

$$\begin{aligned} \text{LoRA}(Q) &= XW^Q + XB^QA^Q \\ \text{LoRA}(K) &= XW^K + XB^KA^K \end{aligned} \tag{3}$$

where A^Q, A^K are the trainable low-rank matrices, and B^Q, and B^K are the projection matrices that project the input X into a lower-dimensional space. The addition of these low-rank matrices to the original query and key matrices allows the model to fine-tune more effectively and efficiently with fewer trainable parameters. Besides one-stage fine-tuning, LLM4TS [Chang *et al.*, 2023] and aLLM4TS [Bian *et al.*, 2024] propose a two-stage fine-tuning strategy to accommodate LLMs to time series data. The pretraining stage is supervised autoregressive fine-tuning, where the backbone predicts contiguous patches based on a sequence of patches as inputs.

3.5 Integrating LLMs in Time Series Models

Rather than directly querying or fine-tuning time series LLMs to generate output, some studies use frozen LLMs as a component that is inherently capable of enhancing the feature space of time series. A frozen LLM can serve as a highly capable function in multi-stage modeling that provides intermediate processing of data or the output of the preceding component, and feeds it to the subsequent neural networks [Xue *et al.*, 2022a; Shi *et al.*, 2023] or regression analysis [Lopez-Lira and Tang, 2023]. Specifically, LLMs can be efficiently applied within a multimodal self-supervised framework for time series analysis. Here, embeddings from time series data and LLM-generated text embeddings are used as positive and negative pairs to refine the model through contrastive loss optimization [Sun *et al.*, 2024; Li *et al.*, 2023]. Because of LLMs' inherent capability to understand natural language, they are also a good fit for generating complex inter-series dependencies for downstream multivariate time series modeling whenever external related text is available. LA-GCN [Xu *et al.*, 2023] and Chen *et al.* [2023b] use LLMs to learn the topological structure of multivariate time series from domain-specific text.

4 Applications of Time Series LLMs

In this section, we review the existing applications of LLMs to general and spatial-temporal time series data, which covers universal and domain-specific areas including finance, transportation, healthcare, and computer vision.

4.1 General Time Series Analysis

Universal Applications

The aforementioned time series LLMs have been evaluated on a wide spectrum of benchmark datasets covering energy, traffic, electricity, weather, illness, business, aeronautics, and security [Zhou *et al.*, 2023; Sun *et al.*, 2024; Gruver *et al.*, 2023; Xue and Salim, 2023; Cao *et al.*, 2024; Chang *et al.*, 2023; Spathis and Kawsar, 2023; Jin *et al.*, 2024; Pan *et al.*, 2024; Liu *et al.*, 2023b; Ansari *et al.*, 2024; Bian *et al.*, 2024]. The tasks include forecasting, classification, imputation, and anomaly detection. These universal modeling methods can be tailored to each of these domains with specific knowledge.

While these applications are designed for structured time series data, a few recent studies have explored LLMs for a type of naturally observed temporal data with irregularities - the event sequence data. LAMP [Shi *et al.*, 2023] first proposes to integrate an event prediction model with an LLM that performs abductive reasoning on real-world events. In the proposed framework, event candidate predictions are generated from historical event data (time, subject, and object) using a pre-trained base event sequence model, and an LLM is prompted to suggest possible cause events. This step is instruction-tuned with a few expert-annotated examples. For retrieval of relevant events, these events will be constructed as embeddings and matched against past events based on cosine similarity scores. Finally, an energy function with a continuous-time Transformer [Xue *et al.*, 2022b] learns to rank predictions with scores and output the event with the strongest retrieved evidence. The proposed framework outperforms state-of-the-art event sequence models on real-world benchmarks, indicating the superior performance of event reasoning via LLMs. Similarly, Gunjal and Durrett [2023] attempts to use an LLM to construct

Method	Data Type	Domain	Task	Modeling Strategy					LLM	Code
				Query	Token	Prompt	Fine-tune	Integrate		
Time-LLM [Jin *et al.*, 2024]	M-TS	General	Forecasting	✗	✓	✓	✗	✗	LLaMA, GPT-2	Yes[1]
OFA [Zhou *et al.*, 2023]	TS	General	Multiple	✗	✓	✗	✓	✗	GPT-2	Yes[2]
TEMPO [Cao *et al.*, 2024]	TS	General	Forecasting	✗	✓	✓	✓	✗	GPT-2	Yes[3]
TEST [Sun *et al.*, 2024]	M-TS	General	Forecasting Classification	✗	✓	✓	✗	✓	BERT, GPT-2 ChatGLM, LLaMA2	Yes[4]
LLM4TS [Chang *et al.*, 2023]	TS	General	Forecasting	✗	✓	✗	✓	✗	GPT-2	No
PromptCast [Xue and Salim, 2023]	TS	General	Forecasting	✓	✗	✓	✗	✗	Bart, BERT, *etc.*	Yes[5]
LLMTIME [Gruver *et al.*, 2023]	TS	General	Forecasting	✓	✓	✗	✗	✗	GPT-3, LLaMA-2	Yes[6]
UniTime [Liu *et al.*, 2023b]	M-TS	General	Forecasting	✗	✓	✓	✓	✓	GPT-2	Yes[7]
aLLM4TS [Bian *et al.*, 2024]	TS	General	Multiple	✗	✓	✗	✓	✗	GPT-2	No
GPT4MTS [Jia *et al.*, 2024]	M-TS	General	Forecasting	✗	✓	✓	✓	✗	GPT-2	No
Chronos [Ansari *et al.*, 2024]	TS	General	Forecasting	✗	✓	✗	✓	✗	GPT-2,T5	Yes[8]
S^2IP-LLM [Pan *et al.*, 2024]	TS	General	Forecasting	✗	✓	✓	✓	✗	GPT-2	Yes[9]
LAMP [Shi *et al.*, 2023]	TS	General	Event Prediction	✓	✗	✓	✗	✓	GPT-3&3.5, LLaMA-2	Yes[10]
[Gunjal and Durrett, 2023]	TS	General	Event Prediction	✓	✗	✓	✗	✗	GPT-3.5, Flan-T5, *etc.*	No
[Yu *et al.*, 2023]	M-TS	Finance	Forecasting	✓	✗	✓	✓	✗	GPT-4, Open-LLaMA	No
[Lopez-Lira and Tang, 2023]	M-TS	Finance	Forecasting	✓	✗	✓	✗	✓	ChatGPT	No
[Xie *et al.*, 2023]	M-TS	Finance	Classification	✓	✗	✓	✗	✗	ChatGPT	No
[Chen *et al.*, 2023b]	M-TS	Finance	Classification	✗	✗	✓	✗	✓	ChatGPT	Partial[11]
METS [Li *et al.*, 2023]	M-TS	Healthcare	Classification	✓	✗	✓	✗	✓	ClinicalBERT	No
[Jiang *et al.*, 2023]	M-TS	Healthcare	Classification	✗	✗	✗	✓	✗	NYUTron(BERT)	Yes[12]
[Liu *et al.*, 2023a]	M-TS	Healthcare	Forecasting Classification	✓	✗	✓	✓	✗	PaLM	No
AuxMobLCast [Xue *et al.*, 2022a]	ST	Mobility	Forecasting	✗	✗	✓	✓	✓	BERT, RoBERTa GPT-2, XLNet	Yes[13]
LLM-Mob [Wang *et al.*, 2023]	ST	Mobility	Forecasting	✓	✗	✓	✗	✗	GPT-3.5	Yes[14]
ST-LLM [Liu *et al.*, 2024]	ST	Traffic	Forecasting	✗	✓	✗	✓	✗	LLaMA, GPT-2	Yes[15]
GATGPT [Chen *et al.*, 2023a]	ST	Traffic	Imputation	✗	✓	✗	✓	✗	GPT-2	No
LA-GCN [Xu *et al.*, 2023]	M-ST	Vision	Classification	✗	✓	✗	✗	✓	BERT	Yes[16]

Table 1: Taxonomy of time series LLMs. The data type **TS** denotes general time series, **ST** denotes spatial-temporal time series, the prefix **M-** indicates multi-modal inputs. The task entry **Multiple** includes forecasting, classification, imputation and anomaly detection. **Query** denotes direct query the whole LLMs for output, **Token** denotes the design of time series tokenization, **Prompt** indicates the design of textual or parameterized time series prompts, **Fine-tune** indicates if the parameters of LLMs are updated, **Integrate** indicates if LLMs are integrated as part of final model for downstream tasks. Code availability is assessed on May 20th, 2024. The Github links are embedded.

structured representations of event knowledge (schema) directly in natural language, to achieve high recall over a set of human-curated events. In the experiments, multiple LLMs are considered and schemas are evaluated on different datasets, which highlights the importance of designing complex prompts for higher event coverage.

Finance

A recent trend in existing literature highlights the emergence of LLMs specialized for financial applications. Yu *et al.* [2023] focuses on a stock return prediction task by incorporating multi-modal data including the historical stock price, generated company profiles, and summarized weekly top news from GPT-4. Based on the designed prompt, this paper tests both instruction-based zero-shot/few-shot queries (with an effective alternative using COT approach) on GPT-4 and instruction-based fine-tuning on Open LLaMA. Results demonstrate that fine-tuned LLMs are capable of making decisions by analyzing multi-modal financial data, thereby extracting meaningful insights and yielding explainable forecasts. Similarly, Lopez-Lira and Tang [2023] directly queries ChatGPT and other large language models for stock market return predictions by using news headlines. A linear regression of the next day's stock return is conducted on the recommendation score. A positive correlation between the scores and subsequent returns is observed, showing the potential of LLMs to comprehend and forecast financial time series.

Xie *et al.* [2023] conducts an extensive study that queries ChatGPT (with designed prompts and COT alternatives) to test its zero-shot capabilities for multi-modal stock movement prediction. The experiments are conducted on three benchmark datasets that contain both stock prices and tweet

data. Interestingly, even if ChatGPT demonstrates its effectiveness, the performance varies across datasets and underperforms even simple traditional methods. Observed limitations suggest the need for specialized fine-tuning techniques (*e.g.,* the aforementioned Yu *et al.* [2023]) in the financial context.

Beyond direct queries or fine-tuning, Chen *et al.* [2023b] proposes a framework that uses an external LLMs as a feature enhancement module for multi-modal stock movement prediction. Specifically, ChatGPT first generates an evolving graph structure of companies via prompting of news headlines at each time step, after which the static features of companies, the inferred structure, and historical stock prices are fed into a GNN and an LSTM for price movement prediction. This paper also provides an evaluation of portfolio performance with higher annualized cumulative returns, lower volatility, and lower maximum drawdown, suggesting the efficacy of LLMs in financial applications.

We also acknowledge recent research efforts to develop financial LLMs where text input includes temporal information [Wu *et al.*, 2023; Xue *et al.*, 2023; Zhang *et al.*, 2023a]. These models, however, are more NLP-centric, with tasks including financial sentiment analysis, Q&A, and named entity recognition, which are less relevant to our survey.

Healthcare

Recent studies in healthcare have highlighted LLMs' capability to comprehend multi-modal medical context including physiological and behavioral time series, such as EEG (Electroencephalogram), ECG (Electrocardiogram), and Electronic Health Records (EHRs). METS [Li *et al.*, 2023] framework aims to integrate LLMs into an ECG encoder classification. The model contains an ECG encoder based on ResNet1d-18, and a frozen large clinical language model, ClinicalBert [Huang *et al.*, 2020] that is pre-trained on all text from the MIMIC III dataset. A multi-modal self-supervised learning framework is used to align the paired ECG and text reports from the same patient while contrasting the unpaired ones via cosine similarity. In the zero-shot testing stage, the medical diagnostic statements constructed from discrete ECG labels are fed to ClinicalBert, and the similarity between ECG embedding and text embedding will be used for ECG classification. It first demonstrates the effectiveness of LLM-based self-supervised learning in multi-modality medical contexts.

Jiang *et al.* [2023] proposes to develop an all-purpose clinical LLM, *i.e.*, NYUTron, which is trained on EHRs and subsequently fine-tuned on three common clinical tasks and two operational tasks, such as the prediction of readmission, in-hospital mortality, comorbidity index, length of stay, as well as the insurance denial status. In this framework, clinical notes and task-specific labels are queried from the NYU Langone EHR database, which are used to pre-train a BERT model with masked language modeling objectives and perform subsequent fine-tuning. The trained model demonstrates improvements compared to traditional benchmarks on all tasks, suggesting the generalization capability of LLMs trained on clinical text. Note that a similar work [Yang *et al.*, 2022] also aims to build a large clinical language model from scratch, but is more tailored to clinical NLP tasks.

In addition to the developments of healthcare LLMs, healthcare datasets have given us other insights. For example, Liu *et al.* [2023a] tests a pre-trained PaLM [Chowdhery *et al.*, 2023] on wearable and medical sensor recordings with three settings, zero-shot, prompt engineering, and prompt tuning for multiple healthcare tasks (cardiac signal analysis, physical activity recognition, metabolic calculation, and mental health). Their results emphasize the importance of healthcare time series for improving the capability of medical language models in the few-shot setting. Similarly, Spathis and Kawsar [2023] provide a case study of the tokenization of popular LLMs on mobile health sensing data, where a modality gap and potential solutions are discussed, such as prompt tuning, model grafting that maps time series via trained encoders onto the same token embedding space as text, as well as the design of new tokenizers for multi-modal time series.

4.2 Spatial-Temporal Time Series Analysis

Traffic

In ST-LLM [Liu *et al.*, 2024], a spatial-temporal tokenization component is proposed so that an LLM is tailored for traffic forecasting tasks, where the input with exogenous information (*e.g.,* hour of day, day of week) is encoded and integrated through point-wise convolutions and linear projections. Furthermore, the partial frozen training strategy is leveraged, with the multi-head attention in the last a few layers unfrozen in the fine-tune process to effectively handle spatial-temporal dependencies. Besides the general setting, the ST-LLM demonstrates advantages in terms of few-shot and zero-shot forecasting scenarios. In addition to forecasting, the spatial-temporal imputation is initially explored by GATGPT [Chen *et al.*, 2023a] that leverages a given topology of traffic networks together with LLMs. It exploits a trainable graph attention module to enhance the embedding of irregular spatial-temporal inputs for imputation tasks.

Human Mobility

Xue *et al.* [2022a] first leverages a non-numerical paradigm to perform spatial-temporal forecasting on human mobility data. Specifically, a mobility prompt, consisting of contextual Place-of-Interest (POI), temporal information, and mobility data, is designed and used to query the pre-trained LLM encoder, based on which the prompt embedding and the numerical token of mobility data is used to fine-tune the decoder to generate the token of prediction. An auxiliary POI category classification task built on top of a fully connected layer helps regularize the model training toward contextual forecasting and improve performance. Instead of using prompt embedding as feature enhancement, LLM-Mob [Wang *et al.*, 2023] directly queries an LLM for not only human movement prediction but also explanations based on an elaborated prompt. It integrates domain-specific knowledge of both long-term (*i.e.*, historical stays) and short-term mobility patterns (*i.e.*, the most recent movements) into the design of context-inclusive prompts. LLMs are guided to comprehend the underlying context of mobility data, and generate accurate forecasts as well as reasonable explanations.

Computer Vision

One recent study of skeleton-based action recognition [Xu *et al.*, 2023] also exhibits the importance of LLMs as an effec-

tive feature enhancement method in computer vision. Motivated by the potential of LLMs to capture the underlying knowledge, provide reasoning, and analyze actions within a skeleton sequence, this paper integrates LLM to generate faithful structural priors and action relations to assist spatial-temporal modeling. The names of all joints and action labels are fed into a pre-trained BERT to get the text embeddings, where edges of skeleton topology and action relation can be calculated using Euclidean distance. The semantic relationships encoded by the LLM can enhance spatial-temporal modeling with graph convolution and facilitate classification.

5 Future Research Opportunities

Time series analysis with LLMs is an emerging and rapidly growing research area. Despite significant advances that have been made in the area, there are still many challenges, which open up a number of research opportunities:

Tokenization & Prompt Design. Tokenization plays a foundational role in capturing temporal dynamics of the input time series data. Existing techniques either rely on a single timestamp or patches of time steps to perform tokenization which could be insufficient to encode the time series for either general or specific applications. Therefore, it is important to develop novel tokenization methods that can better capture the temporal dynamics and facilitate underlying applications. For instance, [Rasul *et al.*, 2023] employed lag features where the lags are derived from a set of appropriate lag indices for quarterly, monthly, weekly, daily, hourly, and second-level frequencies that correspond to the frequencies in the corpus of time series data. Based on appropriate tokenization, it is equally important to investigate how to design better prompts to further improve the model performance. For instance, we may develop a prompt learning architecture based on [Wang *et al.*, 2022] to tailor prompts for specific tasks.

Interpretability. Existing methods for LLMs based time series analysis aim to develop better tokenization, prompt design, fine-tune strategy, and integrate them to improve the model performance. However, these models are typically black-box, and therefore their output lacks explainability. In some applications, it is critical to explain the rationale of the model output to make it trustworthy. For this purpose, we may explore prototype-based methods and gradient-based methods to provide interpretations for the LLMs' output. We may also leverage knowledge distillation to train an explainable student model [Mai *et al.*, 2023a] to enhance the interpretability of LLMs.

Multi-modality. Time series data could be associated with data from other sources. For instance, in healthcare, we may not only collect the continuously monitored heart rate and blood pressure (time series) and medical records (texts and tabular data) but also collect X-rays (images). In this case, it is important to investigate how to incorporate multi-modality input via LLMs, align different modalities of input in the embedding space, and interpret the output accordingly.

Domain Generalization. One of the key challenges for LLM-based time series analysis is domain generalization which aims to generalize the model learned from one or more source domains to unseen target domains. Therefore, it is essential to tackle the distribution shift or domain shift problem by leveraging appropriate time series augmentation techniques, learning temporal features that are invariant across domains (*i.e.*, shared temporal dynamics or structures that are common to all domains), or meta-learning which aims to rapidly adapt to new time series tasks with limited examples from the target domain.

Scaling Laws of Time Series LLMs. One critical research direction over LLMs is to understand their scaling laws, which aim to learn the patterns that depict how the increment of LLMs' size (*e.g.*, in terms of the number of parameters) may impact their performance. Based on time series data, it is also crucial to verify whether the existing scaling laws are still valid based on either zero-shot learning, prompt learning, fine-tuning LLMs, or integration which will be tailored to specific time series tasks and applications.

Time Series LLMs as Agents. LLM-based time series analysis can capture the temporal dynamics of the input time series and therefore can be used to assist in decision-making processes. By analyzing large volumes of time series data and their associated actions or rewards, LLMs can predict the outcomes based on historical data and summarize potential options based on the current status. As agents, time series LLMs can be adapted based on user preference, history, or context, to provide more personalized prediction and decisions. They can also serve as intermediaries or facilitators to seamlessly integrate with various systems and data sources to gather pertinent information, initiate actions, and deliver more extensive services.

Bias and Safety. LLMs are trained on large-scale datasets collected from the internet and other sources which could inevitably involve bias. Because of this, LLMs may not only replicate but also amplify biases. To mitigate this issue, we should consider including a diverse range of data in the training set to reduce potential biases. We may also develop algorithms to detect, assess, and correct potential biases in the LLMs' output. Meanwhile, it is critical for time series LLMs to provide accurate and reliable output, especially in mission-critical systems such as healthcare and power systems. We should conduct rigorous tests over a wide range of scenarios to ensure the reliability and safety of LLMs' outputs.

6 Conclusion

In this survey, we provide a detailed overview of existing time series LLMs. We categorize and summarize the existing methods based on the proposed taxonomy of methodology. We also thoroughly discuss the applications of time series LLMs and highlight future research opportunities.

Acknowledgments

This project was supported by the National Science Foundation (NSF) Grant No. 2338878 and a Research Gift from Morgan Stanley. Yushan Jiang and Zijie Pan were partially funded by the General Electric Fellowship.

Contribution Statement

Yushan Jiang and Zijie Pan are equally contributed. Yuriy Nevmyvaka and Dongjin Song are the correponding authors.

References

[Achiam and et al., 2023] OpenAI Josh Achiam and Steven Adler et al. GPT-4 Technical Report. 2023.

[Ansari *et al.*, 2024] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, et al. Chronos: Learning the language of time series. *arXiv:2403.07815*, 2024.

[Arisoy *et al.*, 2012] Ebru Arisoy, Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Deep Neural Network Language Models. In *NAACL-HLT 2012 Workshop*, 2012.

[Bian *et al.*, 2024] Yuxuan Bian, Xuan Ju, Jiangtong Li, Zhijian Xu, Dawei Cheng, and Qiang Xu. Multi-Patch Prediction: Adapting LLMs for Time Series Representation Learning. *arXiv:2402.04852*, 2024.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, et al. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

[Cao *et al.*, 2024] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting. In *ICLR*, 2024.

[Chang *et al.*, 2023] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. LLM4TS: Aligning Pre-Trained LLMs as Data-Efficient Time-Series Forecasters. *arXiv:2308.08469*, 2023.

[Chen *et al.*, 2023a] Yakun Chen, Xianzhi Wang, and Guandong Xu. GATGPT: A Pre-trained Large Language Model with Graph Attention Network for Spatiotemporal Imputation. *arXiv:2311.14332*, 2023.

[Chen *et al.*, 2023b] Zihan Chen, Lei Nico Zheng, Cheng Lu, Jialu Yuan, and Di Zhu. ChatGPT Informed Graph Neural Network for Stock Movement Prediction. *KDD 2023 Workshop*, 2023.

[Chowdhery *et al.*, 2023] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, et al. PaLM: Scaling Language Modeling with Pathways. *JMLR*, 2023.

[Chung *et al.*, 2022] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, et al. Scaling Instruction-Finetuned Language Models. *arXiv:2210.11416*, 2022.

[Cleveland *et al.*, 1990] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *J. Off. Stat*, 6(1):3–73, 1990.

[Computer, 2023] Together Computer. RedPajama-Data: An Open Source Recipe to Reproduce LLaMA training dataset. 2023.

[Deldari *et al.*, 2022] Shohreh Deldari, Hao Xue, Aaqib Saeed, Jiayuan He, Daniel V Smith, and Flora D Salim. Beyond Just Vision: A Review on Self-Supervised Representation Learning on Multimodal and Temporal Data. *arXiv:2206.02353*, 2022.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ICLR*, 2018.

[Du *et al.*, 2022] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, et al. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *ACL*, 2022.

[Garza and Mergenthaler-Canseco, 2023] Azul Garza and Max Mergenthaler-Canseco. TimeGPT-1. *arXiv:2310.03589*, 2023.

[Geng and Liu, 2023] Xinyang Geng and Hao Liu. OpenLLaMA: An Open Reproduction of LLaMA. 2023.

[Gruver *et al.*, 2023] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large Language Models Are Zero Shot Time Series Forecasters. In *NeurIPS*, 2023.

[Gunjal and Durrett, 2023] Anisha Gunjal and Greg Durrett. Drafting Event Schemas using Language Models. *arXiv:2305.14847*, 2023.

[Houlsby *et al.*, 2019] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *ICML*, 2019.

[Hu *et al.*, 2021] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, et al. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2021.

[Huang *et al.*, 2020] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. *CHIL-2020 Workshop*, 2020.

[Jia *et al.*, 2024] Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. GPT4MTS: Prompt-based Large Language Model for Multimodal Time-series Forecasting. In *AAAI*, 2024.

[Jiang *et al.*, 2023] Lavender Yao Jiang, Xujin Chris Liu, Nima Pour Nejatian, Mustafa Nasir-Moin, Duo Wang, Anas Abidin, et al. Health system-scale language models are all-purpose prediction engines. *Nature*, pages 1–6, 2023.

[Jin *et al.*, 2023] Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, et al. Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook. *arXiv:2310.10196*, 2023.

[Jin *et al.*, 2024] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *ICLR*, 2024.

[Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *ICLR*, 2021.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. *arXiv:2104.08691*, 2021.

[Li *et al.*, 2023] Jun Li, Che Liu, Sibo Cheng, Rossella Arcucci, and Shenda Hong. Frozen Language Model Helps ECG Zero-Shot Learning. In *MIDL*, 2023.

[Lightman *et al.*, 2023] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. *arXiv:2305.20050*, 2023.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019.

[Liu *et al.*, 2023a] Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, et al. Large Language Models are Few-Shot Health Learners. *arXiv:2305.15525*, 2023.

[Liu *et al.*, 2023b] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. UniTime: A Language-Empowered Unified Model for Cross-Domain Time Series Forecasting. *arXiv:2310.09751*, 2023.

[Liu *et al.*, 2024] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-Temporal Large Language Model for Traffic Prediction, 2024.

[Lopez-Lira and Tang, 2023] Alejandro Lopez-Lira and Yuehua Tang. Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models. *arXiv:2304.07619*, 2023.

[Lu *et al.*, 2022] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained Transformers as Universal Computation Engines. In *AAAI*, 2022.

[Ma *et al.*, 2023] Qianli Ma, Zhen Liu, Zhenjing Zheng, Ziyang Huang, Siying Zhu, Zhongzhong Yu, et al. A Survey on Time-Series Pre-Trained Models. *arXiv:2305.10716*, 2023.

[Mai *et al.*, 2023a] C Mai, Y Chang, C Chen, and Z Zheng. Enhanced Scalable Graph Neural Network via Knowledge Distillation. *IEEE TNNLS*, 2023.

[Mai *et al.*, 2023b] Gengchen Mai, Weiming Huang, Jin Sun, Suhang Song, Deepak Mishra, Ninghao Liu, Song Gao, et al. On the Opportunities and Challenges of Foundation Models for Geospatial Artificial Intelligence. *arXiv:2304.06798*, 2023.

[Nie *et al.*, 2023] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*, 2023.

[Pan *et al.*, 2024] Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. S^2IP-LLM: Semantic Space Informed Prompt Learning with LLM for Time Series Forecasting. *arXiv:2403.05798*, 2024.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 2019.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *JMLR*, 2020.

[Rasul *et al.*, 2023] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, et al. Lag-Llama: Towards Foundation Models for Time Series Forecasting, 2023.

[Shi *et al.*, 2023] Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. Language Models Can Improve Event Prediction by Few-Shot Abductive Reasoning. In *NeurIPS*, 2023.

[Spathis and Kawsar, 2023] Dimitris Spathis and Fahim Kawsar. The first step is the hardest: Pitfalls of Representing and Tokenizing Temporal Data for Large Language Models. *arXiv:2309.06236*, 2023.

[Sun *et al.*, 2024] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. TEST: Text Prototype Aligned Embedding to Activate LLM's Ability for Time Series. In *ICLR*, 2024.

[Touvron *et al.*, 2023a] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971*, 2023.

[Touvron *et al.*, 2023b] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv:2307.09288*, 2023.

[Wang *et al.*, 2022] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, et al. Learning to Prompt for Continual Learning. In *CVPR*, 2022.

[Wang *et al.*, 2023] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. Where Would I Go Next? Large Language Models as Human Mobility Predictors, 2023.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*, 2022.

[Wu *et al.*, 2022] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, et al. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*, 2022.

[Wu *et al.*, 2023] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. BloombergGPT: A Large Language Model for Finance. 2023.

[Xie *et al.*, 2023] Qianqian Xie, Weiguang Han, Yanzhao Lai, Min Peng, and Jimin Huang. The Wall Street Neophyte: A Zero-Shot Analysis of ChatGPT Over MultiModal Stock Movement Prediction Challenges. *arXiv:2304.05351*, 2023.

[Xu *et al.*, 2023] Haojun Xu, Yan Gao, Zheng Hui, Jie Li, and Xinbo Gao. Language Knowledge-Assisted Representation Learning for Skeleton-Based Action Recognition. *arXiv:2305.12398*, 2023.

[Xue and Salim, 2023] Hao Xue and Flora D Salim. PromptCast: A New Prompt-based Learning Paradigm for Time Series Forecasting. *IEEE TKDE*, 2023.

[Xue *et al.*, 2022a] Hao Xue, Bhanu Prakash Voutharoja, and Flora D Salim. Leveraging Language Foundation Models for Human Mobility Forecasting. In *SIGSPATIAL*, 2022.

[Xue *et al.*, 2022b] Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. HYPRO: A Hybridly Normalized Probabilistic Model for Long-Horizon Prediction of Event Sequences. *NeurIPS*, 2022.

[Xue *et al.*, 2023] Siqiao Xue, Fan Zhou, Yi Xu, Hongyu Zhao, Shuo Xie, et al. WeaverBird: Empowering Financial Decision-Making with Large Language Model, Knowledge Base, and Search Engine. *arXiv:2308.05361*, 2023.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*, 2019.

[Yang *et al.*, 2022] Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, et al. A large language model for electronic health records. *NPJ Digital Medicine*, 5(1):194, 2022.

[Yin *et al.*, 2023] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A Survey on Multimodal Large Language Models. *arXiv:2306.13549*, 2023.

[Yu *et al.*, 2023] Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal Data Meets LLM–Explainable Financial Time Series Forecasting. *arXiv:2306.11025*, 2023.

[Zhang *et al.*, 2023a] Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. Instruct-FinGPT: Financial Sentiment Analysis by Instruction Tuning of General-Purpose Large Language Models. *arXiv:2306.12659*, 2023.

[Zhang *et al.*, 2023b] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, et al. Multimodal Chain-of-Thought Reasoning in Language Models. *arXiv:2302.00923*, 2023.

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, et al. A Survey of Large Language Models. *arXiv:2303.18223*, 2023.

[Zhou *et al.*, 2023] Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. One Fits All: Power General Time Series Analysis by Pretrained LM. In *NeurIPS*, 2023.