Large Language Models for Time Series: A Survey

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K. Gupta and Jingbo Shang
University of California, San Diego
{xiyuanzh, rrchowdh, rgupta, jshang}@ucsd.edu

Abstract

Large Language Models (LLMs) have seen significant use in domains such as natural language processing and computer vision. Going beyond text, image and graphics, LLMs present a significant potential for analysis of time series data, benefiting domains such as climate, IoT, healthcare, traffic, audio and finance. This survey paper provides an in-depth exploration and a detailed taxonomy of the various methodologies employed to harness the power of LLMs for time series analysis. We address the inherent challenge of bridging the gap between LLMs’ original text data training and the numerical nature of time series data, and explore strategies for transferring and distilling knowledge from LLMs to numerical time series analysis. We detail various methodologies, including (1) direct prompting of LLMs, (2) time series quantization, (3) aligning techniques, (4) utilization of the vision modality as a bridging mechanism, and (5) the combination of LLMs with tools. Additionally, this survey offers a comprehensive overview of the existing multimodal time series and text datasets in diverse domains, and discusses the challenges and future opportunities of this emerging field.

1 Introduction

Time series analysis plays a critical role in a variety of fields, including climate modeling, traffic management, healthcare monitoring and finance analytics. Time series analysis comprises a wide range of tasks such as classification [Liu et al., 2023b], forecasting [Gruver et al., 2023], anomaly detection, and imputation. Traditionally, these tasks have been tackled using classical signal processing techniques such as time-frequency analysis and decomposition-based approaches. More recently, deep learning approaches like Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs) [Zhang et al., 2023a], and Transformers [Jin et al., 2023a] have revolutionized this field and proved effective in extracting meaningful patterns from time series data, making them the primary approaches of time series analysis in various application domains.

In recent years, Large Language Models (LLMs) have gained substantial attention particularly in the fields of Natural Language Processing (NLP) and Computer Vision (CV). Prominent models such as GPT-4 have transformed the landscape of text processing by offering unprecedented accuracy in tasks such as text generation, translation, sentiment analysis, question answering and summarization. In the CV domain, Large Multimodal Models (LMMs) have also facilitated advancements in image recognition, object detection, and generative tasks, leading to more intelligent and capable visual systems [Girdhar et al., 2023]. Inspired by these successes, researchers are now exploring the potential of LLMs in the realm of time series analysis, expecting further breakthroughs, as shown in Figure 1. While several surveys offer a broad perspective on large models for time series in general [Jin et al., 2023b; Ma et al., 2023], these do not specifically focus on LLMs or the key challenge of bridging modality gap, which stems from LLMs being originally trained on discrete textual data, in contrast to the continuous numerical nature of time series.

Our survey uniquely contributes to the existing literature by emphasizing how to bridge such modality gap and transfer knowledge from LLMs for time series analysis. Our survey also covers more diverse application domains, ranging from climate, Internet of Things (IoT), to healthcare, traffic management, and finance. Moreover, certain intrinsic properties of time series, like continuity, auto-regressiveness, and dependency on the sampling rate, are also shared by audio, speech, and music data. Therefore, we also present representative LLM-based works from these domains to explore how we can use LLMs for other types of time series. We present
a comprehensive taxonomy by categorizing these methodolo-
gies into five distinct groups, as shown in Figure 2. If we
outline typical LLM-driven NLP pipelines in five stages - in-
put text, tokenization, embedding, LLM, output - then each
category of our taxonomy targets one specific stage in this
pipeline. Specifically, (i) Prompting (input stage) treats time
series data as raw text and directly prompts LLMs with time
series; (ii) Time Series Quantization (tokenization stage) dis-
cretizes time series as special tokens for LLMs to process;
(iii) Aligning (embedding stage) designs time series encoder
to align time series embeddings with language space; (iv) Vi-
sion as Bridge (LLM stage) connects time series with Vision-
Language Models (VLM) by employing visual representa-
tions as a bridge; (v) Tool Integration (output stage) adopts
LLMs to output tools to benefit time series analysis. Beyond
this taxonomy, our survey also compiles an extensive list of
existing multimodal datasets that incorporate both time series
and text. We conclude our paper by discussing future research
directions in this emerging and promising field.

We maintain an up-to-date Github repository\footnote{\url{https://github.com/xiuyanzh/awesome-llm-time-series}} which in-
cludes all the papers and datasets discussed in the survey.

\section{Background and Problem Formulation}

Large language models are characterized by their vast num-
er of parameters and extensive training data. They excel in
understanding, generating, and interpreting human lan-
guage, and recently represent a significant advancement in
artificial intelligence. The inception of LLMs can be traced
back to models like GPT-2, BERT, BART, and T5, which
laid the foundational architecture. Over time, the evolution
of these models has been marked by increasing complex-
ity and capabilities, such as LLLAMA-2, PaLM, and GPT-4.
More recently, researchers have developed multimodal large
language models to integrate and interpret multiple forms of
data, such as text, images, and time series, to achieve a more
comprehensive understanding of information.

\section{Taxonomy}

This survey focuses on how LLMs could benefit time series
analysis. We first define the mathematical formulation for the
input and output, which may contain time series or (and) text
depending on the downstream tasks, as well as the models.

\textbf{Input.} Denoted as $x$, composed of time series $x_s \in \mathbb{R}^{T \times c}$ and optional text data $x_t$ represented as strings, where $T, c$
represent the sequence length and the number of features.

\textbf{Output.} Denoted as $y$ and may represent time series, text
or numbers depending on the specific downstream task. For
time series generation or forecasting task, $y$ represents gen-
erated time series $y_s$ or predicted $k$-step future time series
$y_s^{T+1:T+k}$. For text generation task, such as report genera-
tion, $y$ represents text data $y_t$. For time series classification
or regression task, $y$ represents numbers indicating the pre-
dicted classes or numerical values.

\textbf{Model.} We use $f_\theta$ parameterized by $\theta$, $g_\phi$ parameterized by
\phi, and $h_\psi$ parameterized by $\psi$ to represent language, time se-
ries and vision models, where $f_\theta$ is typically initialized from
pre-trained large language models. We optimize parameters
$\theta, \phi$ and $\psi$ through loss function $L$.

\section{3 Taxonomy}

In this section, we detail our taxonomy of applying LLMs
for time series analysis, categorized by five groups. We sum-
marize the representative works, mathematical formulation,
advantages and limitations of each category in Table 1.

### 3.1 Prompting

\textbf{Number-Agnostic Tokenization.} The method treats
numerical time series as raw textual data and directly prompts
existing LLMs. For example, PromptCast \cite{xue2022promptcasting} proposes prompt-based time series forecasting by con-
verting numerical time series into text prompts and forecast-
ing time series in a sentence-to-sentence manner. The input
prompts are composed of context and questions following
pre-defined templates, e.g., “From $\{t_1\}$ to $\{t_{\text{end}}\}$, the aver-
age temperature of region $\{U_{m}\}$ was $\{x_{m}^{T}\}$ degree on each

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{taxonomy.png}
\caption{Left: Taxonomy of LLMs for time series analysis (prompting, quantization, aligning which is further categorized into two groups as detailed in Figure 4, vision as bridge, tool integration). For each category, key distinctions are drawn in comparison to the standard LLM pipeline shown at the top of the figure. Right: We present representative works for each category, sorted by their publication dates. The use
of arrows indicates that later works build upon earlier studies. Dark(light)-colored boxes represent billion(million)-parameter models. Icons to the left of the text boxes represent the application domains of domain-specific models, with icons’ meanings illustrated in Figure 1.}
\end{figure}
day. What is the temperature going to be on \{t_{\text{obs}}\}?” Similar prompting methods have been applied to forecast Place-of-Interest (POI) customer flows (AuxMobLCast) and user’s next location (LLM-Mob). Recent works also prompt PaLM-24B for health-related tasks such as activity recognition and daily stress estimate [Liu et al., 2023b]. For example, they prompt the model to “classify the following accelerometer data in meters per second squared as either walking or running: 0.052, 0.052, 0.052, 0.051, 0.052, 0.055, 0.051, 0.056, 0.06, 0.064”. Other examples include extracting historical price features such as open, close, high, and low prices to prompt ChatGPT in a zero-shot fashion [Xie et al., 2023a].

Number-Specific Tokenization. More recently, LLM-Time [Gruver et al., 2023] pointed out that Byte Pair Encoding (BPE) tokenization has the limitation of breaking a single number into tokens that don’t align with the digits, leading to inconsistent tokenization across different floating point numbers and complicating arithmetic operations. Therefore, following LLMs such as LLaMA and PaLM, they propose to insert spaces between digits to ensure distinct tokenization of each digit and use a comma (“,”) to separate each time step in a time series. They also scale time series to optimize token usage and keep fixed precision (e.g., two digits of precision) to efficiently manage context length. For example, they convert “0.123, 1.23, 12.3, 123.0” to “1 2 3 1 2 3 0, 1 2 3 0”. Meanwhile, BloomberGPT [Wu et al., 2023] trains on financial data with text and numerical data and places each digit in its own chunk to better handle numbers. Using similar space-prefixed tokenization, recent works also show that large language models are general pattern machines capable of sequence transformation, completion and improvement.

3.2 Quantization

Quantization based method converts numerical data into discrete representations as input to LLMs. This approach can be further divided into two main categories based on the discretization technique employed.

Discrete Indices from VQ-VAE. The first type of quantization method transforms continuous time series into discrete indices as tokens. Among them one of the most popular methods is training a Vector Quantized-Variational AutoEncoder (VQ-VAE), which learns a codebook \( \mathcal{C} = \{ \mathbf{c}_i \}_{i=1}^K \) of \( K D \)-dimensional codewords \( \mathbf{c}_i \in \mathbb{R}^D \) to capture the latent representations, as illustrated in Figure 3a. The method identifies the nearest neighbor \( k_i \) of each step \( i \) of the encoded time series representation \( g_{\phi}(\mathbf{x}_i) \in \mathbb{R}^S \times D \) in the codebook \( S \) denotes the cumulative stride of VQ-VAE encoder), and uses the corresponding indices \( k \) as the quantized input to language models:

\[
\mathbf{q}_i = \mathbf{c}_{k_i}, k_i = \arg \min_j \| g_{\phi}(\mathbf{x}_i) - \mathbf{c}_j \|_2, \mathbf{k} = \{ k_i \}_{i=1}^S. \tag{1}
\]

Based on VQ-VAE, Auto-TTE [Chung et al., 2023] quantizes ECGs into discrete formats and generates 12-lead ECG signals conditioned on text reports. DeWave [Duan et al., 2023] adapts VQ-VAE to derive discrete codec encoding and aligns it with pre-trained BART for open-vocabulary EEG-to-text translation tasks. TOTEM [Talukder and Gkioxari, 2023] also quantizes time series through VQ-VAE as input to Transformers for multiple downstream applications such as forecasting, classification, and translation. In the audio domain, UniAudio [Yang et al., 2023] tokenizes different types of target audio using Residual Vector Quantization (RVQ) (a hierarchy of multiple vector quantizers) and supports 11 audio generation tasks. Viola unifies various crossmodal tasks involving speech and text by converting speech utterances to discrete tokens through RVQ. AudioGen learns discrete audio representations using vector quantization layers and generates audio samples conditioned on text inputs.

Discrete Indices from K-Means. Apart from employing VQ-VAE, researchers have also explored K-Means clustering for index-based tokenization, which uses the centroid indices as discretized tokens, as shown in Figure 3b. Such methods are mostly applied in the audio domain. For example, SpeechGPT shows capability to perceive and generate multimodal contents using K-Means based discrete unit extractor. AudioLM discretizes codes produced by a neural audio codec using K-means clustering to achieve high-quality synthesis. It also combines discretized activations of language models pre-trained on audio using RVQ to capture long-term content. Following the same quantization procedure, AudioPaLM [Rubenstein et al., 2023] aligns PaLM-2 and AudioLM with a joint vocabulary that can represent speech and text with discrete tokens.

Discrete Indices from Other Techniques. Apart from the VQ-VAE and K-Means based time-domain quantization, FreqTST [Li et al., 2023] utilizes frequency spectrum as a common dictionary to discretize time series into frequency units with weights for downstream forecasting task. TOTEM [Talukder and Gkioxari, 2023] aligns PaLM-2 and AudioLM with a joint vocabulary that can represent speech and text with discrete tokens.

Text Categories. The second type of quantization converts numerical data into pre-defined text categories, which is primarily adopted in financial domain. For example, TDML [Yu et al., 2023] categorizes the weekly price fluctuations into 12 bins represented as “D1” or “U1”, where “D” indicates a decrease in price and “U” means an increase, and \( i = 1, 2, 3, 4, 5, 5+ \) represents the level of price change.

3.3 Aligning

The third type of works trains a separate encoder for time series, and aligns the encoded time series to the semantic space.
of language models. These works can be further categorized into two groups based on their specific aligning strategies, as illustrated in Figure 4.

**Similarity Matching through Contrastive Loss.** The first type of method aligns the time series embeddings with text embeddings through similarity matching, such as minimizing the contrastive loss:

$$\mathcal{L} = - \frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(g(o(x_{s1})), f_0(x_{t1}))}{\sum_{k=1}^{B} \exp(g(o(x_{s1})), f_0(x_{tk}))}$$

(2)

where $B$, $\gamma$ represent batch size and temperature parameter that controls distribution concentrations, and sim represents similarity score, typically computed as inner product:

$$\text{sim}(g(o(x_{s1})), f_0(x_{t1})) = \langle g(o(x_{s1})), f_0(x_{t1}) \rangle.$$  

(3)

For instance, ETP [Liu et al., 2023a] integrates contrastive learning based pre-training to align electrocardiography (ECG) signals with textual reports. Contrastive framework is also used to align 17 clinical measurements collected in Intensive Care Unit (ICU) to their corresponding clinical notes [King et al., 2023]. TEST [Sun et al., 2023] uses contrastive learning to generate instance-wise, feature-wise, and text-prototype-aligned time series embeddings to align with text embeddings. TENT [Zhou et al., 2023b] aligns text embeddings with IoT sensor signals through a unified semantic space using contrastive learning. JoLT [Cai et al., 2023] utilizes Querying Transformer (Q-Former) optimized with contrastive loss to align time series and text representations.

**Similarity Matching through Other Losses.** Apart from contrastive loss, other loss functions are also employed to optimize similarity matching between time series embeddings and text embeddings. ECG-LLM [Qiu et al., 2023] aligns the distribution between ECG and language embedding from ECG statements with an Optimal Transport based loss function to train an ECG report generation model. MTAM [Han et al., 2022] uses various aligning techniques, such as Canonical Correlation Analysis and Wasserstein Distance, as loss functions to align electroencephalography (EEG) features with their corresponding language descriptions.

**LLMs as Backbones.** The second type of aligning method directly uses large language models as backbones following time series embedding layers. EEG-to-Text [Wang and Ji, 2022] feeds EEG embeddings to pre-trained BART for open vocabulary EEG-To-Text decoding and EEG-based sentiment classification. GPT4TS [Zhou et al., 2023a] uses patching embeddings as input to frozen pre-trained GPT-2 where the positional embedding layers and self-attention blocks are retained during time series fine-tuning. The method provides a unified framework for seven time series tasks, including few-shot or zero-shot learning. Following GPT4TS, researchers further incorporated seasonal-trend decomposition (TEMPO [Cao et al., 2023]), two-stage fine-tuning (LLM4TS [Chang et al., 2023]), domain descriptions (UniTime), graph attention mechanism (GATGPT), and spatial-temporal embedding module (ST-LLM). TimeLLM [Jin et al., 2023a] reprograms time series data into text prototypes as input to LLaMA-7B. It also provides natural language prompts such as domain expert knowledge and task instructions to augment input context. Lag-LLama builds univariate probabilistic time series forecasting model based on LLaMA architecture. In the audio, speech and music domains, researchers have also designed dedicated encoders to embed speech (WavPrompt, Speech LLaMA), music (MULLaMA), and general audio inputs (LTU [Gong et al., 2023], SALMONN [Tang et al., 2023]), and feed the embeddings to large language models.

### 3.4 Vision as Bridge

Time series data can be effectively interpreted or associated with visual representations, which align closer with textual data and have demonstrated successful integrations with large language models. Therefore, researchers have also leveraged vision modality as a bridge to connect time series with LLMs.

**Paired Data.** ImageBind [Girdhar et al., 2023] uses image-paired data to bind six modalities (images, text, audio, depth, thermal, and Inertial Measurement Unit (IMU) time series) and learn a joint embedding space, enabling new emergent alignments and capabilities. PandaGPT [Su et al., 2023] further combines the multimodal encoders from ImageBind and large language models to enable visual and auditory instruction-following capabilities. IMU2CLIP [Moon et al., 2022] aligns IMU time series with video and text, by projecting them into the joint representation space of Contrastive Language-Image Pre-training (CLIP). AnyMAL [Moon et al., 2023] builds upon IMU2CLIP by training a lightweight adapter to project the IMU embeddings into the text token embedding space of LLaMA-2-70B. It is also capable of transforming data from other modalities, such as images, videos, audio, into the same text embedding space.

**Physics Relationships.** IMUGPT [Leng et al., 2023] generates IMU data from ChatGPT-augmented text descriptions. It first generates 3D human motion from text using pre-trained motion synthesis model, and derives IMU data from 3D motion based on physics relationships of motion kinetics.
### Table 1: Summary of five major categories of applying LLMs for time series analysis, including their respective subcategories, representative works, mathematical formulations, advantages and limitations. \( q \) and \( x_t \) represent text-based quantization process and image data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Subcategory</th>
<th>Representative Works</th>
<th>Equations</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompting</td>
<td>Number-Agnostic</td>
<td>PromptCast [Xue and Salim, 2022]</td>
<td>( y = f_\theta(x_i, x_t) )</td>
<td>easy to implement; zero-shot capability</td>
<td>lose semantics; not efficient</td>
</tr>
<tr>
<td></td>
<td>Number-Specific</td>
<td>LLMTIME [Gruver et al., 2023]</td>
<td>( k_i = \arg \min_j | g_\phi(x_j) - c_i |<em>2 ) ( k = { k_i }</em>{i=1}^2 ) ( y = f_\theta(k, x_t) )</td>
<td>flexbility of index and time series conversion</td>
<td>may require two-stage training</td>
</tr>
<tr>
<td>Quantization</td>
<td>VQ-VAE</td>
<td>DeWave [Duan et al., 2023]</td>
<td>( y = f_\theta(q(x_i), x_t) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>K-Means</td>
<td>AudioPALM [Rubenstein et al., 2023]</td>
<td>( L = \sim(g_\phi(x_d), f_\theta(x_t)) ) ( y = f_\psi(g_\phi(x_d), x_t) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aligning</td>
<td>Similarity Match</td>
<td>ETP [Liu et al., 2023a]</td>
<td>( y = g_\phi(x_i) )</td>
<td>align semantics of different modalities;</td>
<td>complicated design and fine-tuning</td>
</tr>
<tr>
<td></td>
<td>LLM Backbone</td>
<td>GPT4TS [Zhou et al., 2023a]</td>
<td>( L = \sim(g_\phi(x_d), h_\psi(x_s)) ) ( y = h_\psi(x_s) )</td>
<td>end-to-end training</td>
<td></td>
</tr>
<tr>
<td>Vision as</td>
<td>Paired Data</td>
<td>ImageBind [Girdhar et al., 2023]</td>
<td>( y = h_\psi(x_s) )</td>
<td>additional visual knowledge for all data</td>
<td>not hold</td>
</tr>
<tr>
<td>Bridge</td>
<td>TS Plots</td>
<td>Tool [Wimmer and Rekabsaz, 2023]</td>
<td>( y = h_\psi(x_s) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>API</td>
<td>ToolLLM [Qin et al., 2023]</td>
<td>( y = z(x_s) )</td>
<td>empower LLM; with more abilities; not end-to-end</td>
<td></td>
</tr>
</tbody>
</table>

#### Time Series Plots as Images.
CLIP-LSTM [Wimmer and Rekabsaz, 2023] transforms stock market data into sequences of texts and images of price charts, and leverages pre-trained CLIP vision-language model to generate features for downstream forecasting. Insight Miner [Zhang et al., 2023b] converts time series windows into images using lineplot, and feeds images into vision language model LLaVA to generate time series trend descriptions.

#### 3.5 Tool
This type of method does not directly use large language models to process time series. Instead, it applies large language models to generate indirect tools \( z(\cdot) \), such as code and API calls, to benefit time series related tasks.

- **Code.** CTG++ [Zhong et al., 2023] applies GPT-4 to generate differentiable loss functions in a code format from text descriptions to guide the diffusion model to generate traffic trajectories. With this two-step translation, the large language model and diffusion model efficiently bridge the gap between user intent and traffic simulation.

- **API Call.** ToolLLM [Qin et al., 2023] introduces a general tool-use framework composed of data construction, model training, and evaluation. This framework includes API calls for time series tasks such as weather and stock forecasting.

#### Text Domain Knowledge.
SHARE [Zhang et al., 2023a] exploits the shared structures in human activity label names and proposes a sequence-to-sequence structure to generate label names as token sequences to preserve the shared label structures. It applies GPT-4 to augment semantics of label names. GG-LLM [Graule and Isler, 2023] leverages LLaMA-2 to encode world knowledge of common human behavioral patterns to predict human actions without further training. SCRL-LG [Ding et al., 2023] leverages LLaMA-7B as stock feature selectors to extract meaningful representations from news headlines, which are subsequently employed in reinforcement learning for precise feature alignments.

#### 4 Comparison within the Taxonomy
We compare the five categories of our taxonomy and provide general guidelines for which category to choose based on considerations of data, model, efficiency and optimization.

**Data.** When no training data is available and the objective is to apply LLM for time series in an zero-shot fashion, it is preferable to use prompting-based methods. This is because direct prompting enables the utilization of pre-trained language models’ inherent capabilities without fine-tuning. However, representing numbers as strings can diminish the semantic value intrinsically tied to numerical data. Therefore, with adequate training data, quantization or aligning-based methods become more advantageous. As shown in Figure 2, these two categories are the most extensively studied ones in existing literature. Furthermore, if time series data can be interpreted or associated with visual representations, these representations can be incorporated to utilize the intrinsic knowledge embedded in the vision modality or pre-trained vision-language models.

**Model.** Prompting and tool integration methods tend to apply billion-parameter models as they often apply off-the-self LLMs without architectural modifications. By contrast, aligning and quantization methods vary from million parameter diverse architectures.

**Efficiency.** Prompting-based methods are not efficient for numerical data with high precision, as well as for multivariate time series as it requires transforming each dimension into separate univariate time series, resulting in extremely long input. They are also less efficient for long-term predictions due to the computational demands of generating long sequences. These methods are more effective when dealing with simple numerical data that is richly interwoven with textual information, such as opening and closing stock prices in financial news articles. By contrast, quantization and aligning meth-
ods are more efficient to handle long sequences, as time series are typically down-sampled or segmented into patches before feeding into large language models.

Optimization. Depending on the specific discretization technique, quantization based method may require a two-stage training process (such as first training the VQ-VAE model), which may result in sub-optimal performance compared to that achieved through end-to-end training in aligning methods. Using large language models as indirect tools empowers LLMs with more capabilities to manage numerical data, but also raises the level of complexity to optimize both LLMs and other components in an end-to-end fashion. Therefore, existing works of tool integration typically employ off-the-shelf LLMs without further fine-tuning.

5 Multimodal Datasets

Applying LLMs for time series benefits from the availability of multimodal time series and text data. In this section, we introduce representative multimodal datasets organized by their respective domains (Table 2). Due to space limit, additional datasets are listed in our Github repository.

Internet of Things (IoT). Human activity recognition is an important task in IoT domain, which identifies human activities given time series collected with IoT devices (such as IMU sensors). The corresponding text data are the labels or text descriptions of these activities. Ego4D [Grauman et al., 2022] presents 3,670 hours of daily-life activity data with multiple modalities, including IMU time series, and dense temporally-aligned textual descriptions of the activities. Ego-Exo4D further offers three kinds of paired natural language datasets including expert commentary, narrate-and-act descriptions provided by the participants themselves, and atomic action descriptions similar as Ego4D. DeepSQA [Xing et al., 2021] presents a generalized Sensory Question Answering (SQA) framework to facilitate querying raw sensory data related to human activities using natural language.

Finance. PIXIU [Xie et al., 2023b] presents multi-task and multi-modal instruction tuning data in the financial domain with 136K data samples. It contains both financial natural language understanding and prediction tasks, and covers 9 datasets of multiple modalities such as text and time series. MoAT [Lee et al., 2023] constructs multimodal datasets with textual information paired with time series for each timestep, such as news articles extracted with relevant keywords, mostly covering finance related domains such as fuel, metal, stock and bitcoin.

Healthcare. Zuco datasets [Hollenstein et al., 2019] contain simultaneous eye-tracking and EEG during natural reading and during annotation. PTB-XL [Wagner et al., 2020] offers comprehensive metadata regarding ECG annotated by expert cardiologists, covering information such as ECG reports, diagnostic statements, diagnosis likelihoods, and signal-specific properties. Based on PTB-XL, ECG-QA [Oh et al., 2023] introduces the first Question Answering dataset for ECG analysis, containing 70 question templates that cover a wide range of clinically relevant ECG topics.

Audio/Music/Speech. AudioSet is a collection of 2 million 10-second audio clips excised from YouTube videos and labeled with the sounds that the clip contains from a set of 527 labels. OpenAQA-5M [Gong et al., 2023] dataset consists of 1.9 million closed-ended and 3.7 million open-ended, diverse (audio, question, answer) tuples. MusicCaps [Agostinelli et al., 2023] is a high-quality music caption dataset, including 5.5K music clips. MTG-Jamendo is a dataset with 55,000 audio songs in various languages. Libri-Light is an English dataset encompassing 60,000 hours of speech data. Common-

<table>
<thead>
<tr>
<th>Domain</th>
<th>Dataset</th>
<th>Size</th>
<th>Major Modalities</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT</td>
<td>Ego4D&lt;sup&gt;7&lt;/sup&gt; [Grauman et al., 2022]</td>
<td>3,670h data, 3.85M narrations</td>
<td>text, IMU, video, audio, 3D</td>
<td>classification, forecasting</td>
</tr>
<tr>
<td></td>
<td>DeepSQA&lt;sup&gt;3&lt;/sup&gt; [Xing et al., 2021]</td>
<td>25h data, 91K questions</td>
<td>text, imu</td>
<td>classification, QA</td>
</tr>
<tr>
<td>Finance</td>
<td>PIXIU&lt;sup&gt;4&lt;/sup&gt; [Xie et al., 2023b]</td>
<td>136K instruction data</td>
<td>text, tables</td>
<td>5 NLP tasks, forecasting</td>
</tr>
<tr>
<td></td>
<td>MoAT&lt;sup&gt;5&lt;/sup&gt; [Lee et al., 2023]</td>
<td>6 datasets, 2K timesteps</td>
<td>text, time series</td>
<td>forecasting</td>
</tr>
<tr>
<td>Healthcare</td>
<td>Zuco 2.0&lt;sup&gt;6&lt;/sup&gt; [Hollenstein et al., 2019]</td>
<td>739 sentences</td>
<td>text, eye-tracking, EEG</td>
<td>classification, text generation</td>
</tr>
<tr>
<td></td>
<td>PTB-XL&lt;sup&gt;7&lt;/sup&gt; [Wagner et al., 2020]</td>
<td>60h data, 71 unique statements</td>
<td>text, ECG</td>
<td>classification</td>
</tr>
<tr>
<td></td>
<td>ECG-QA&lt;sup&gt;8&lt;/sup&gt; [Oh et al., 2023]</td>
<td>70 question templates</td>
<td>text, ECG</td>
<td>classification, QA</td>
</tr>
<tr>
<td>Audio</td>
<td>OpenAQA-5M&lt;sup&gt;9&lt;/sup&gt; [Gong et al., 2023]</td>
<td>5.6M (audio, QA) tuples</td>
<td>text, audio</td>
<td>tagging, classification</td>
</tr>
<tr>
<td>Music</td>
<td>MusicCaps&lt;sup&gt;10&lt;/sup&gt; [Agostinelli et al., 2023]</td>
<td>5.5K music clips</td>
<td>text, music</td>
<td>captioning, generation</td>
</tr>
<tr>
<td>Speech</td>
<td>CommonVoice&lt;sup&gt;11&lt;/sup&gt; [Ardila et al., 2019]</td>
<td>7,335h in 60 languages</td>
<td>text, speech</td>
<td>ASR, translation</td>
</tr>
</tbody>
</table>

Table 2: Summary of representative time series and text multimodal datasets.

<sup>3</sup>https://ego4d-data.org/
<sup>4</sup>https://github.com/nelsi/DeepSQA
<sup>5</sup>https://github.com/chancefocus/PIXIU
<sup>6</sup>https://openreview.net/pdf?id=uRXxnoqDHH
<sup>7</sup>https://osf.io/2urht/
<sup>8</sup>https://physionet.org/content/ptb-xl/1.0.3/
<sup>9</sup>https://github.com/Jwoo5/ecg-qa
<sup>10</sup>https://github.com/YuanGongND/ltu
<sup>11</sup>https://www.kaggle.com/datasets/googleai/musiccaps
<sup>12</sup>https://commonvoice.mozilla.org/en/datasets
<sup>13</sup>https://github.com/xiyuanzh/awesome-llm-time-series
Voice [Ardila et al., 2019] is a multilingual speech dataset consisting of 7,335 validated hours in 60 languages.

These datasets offer valuable benchmarks for multimodal time series and text analysis. These contain both time series focused tasks, including classification, which is evaluated using accuracy and macro-F1 scores, and forecasting, which utilizes metrics such as MSE, MAE, RMSE, and MAPE, as well as NLP focused tasks such as captioning, question answering, and translation, assessed through BLEU, ROUGE, METEOR, and EM scores, among others.

6 Challenges and Future Directions

6.1 Theoretical Understanding

Existing works empirically show the benefits of applying LLMs for time series analysis. For example, recent works have empirically shown that large language models learn linear representations of space and time across multiple scales that are robust to prompting variations. Despite these empirical findings, there remains a gap in theoretical understanding of how models, primarily trained on textual data, can effectively interpret numerical time series. As a preliminary theoretical analysis, it is proved that Transformer models can universally approximate arbitrary continuous sequence-to-sequence functions on a compact domain [Yun et al., 2019]. Additionally, GPT4TS [Zhou et al., 2023a] theoretically shows that such generic capability of large language models can be related to Principal Component Analysis (PCA), as minimizing the gradient with respect to the self-attention layer shares similarities with PCA. Further investigations on the generalizability of large language models on numerical data is essential to establish solid understanding of the synergy between LLMs and time series analysis.

6.2 Multimodal and Multitask Analysis

Existing papers that apply LLMs for time series analysis mostly focus on single modality and single task at a time, such as forecasting, classification, text generation, and do not support simultaneous multimodal and multitask analysis. In computer vision and audio domains, models such as Unified-IO and UniAudio [Yang et al., 2023] have unified multiple input modalities into a sequence of discrete vocabulary tokens to support multiple tasks within a single transformer-based architecture. More research into leveraging LLMs for multimodal and multitask analysis would lead to more powerful time series foundation models.

6.3 Efficient Algorithms

Time series, especially those that are multivariate or possess long history information may increase the computational complexity for existing large language models. Patching (treating each segmented patch as a token) has been a widely adopted strategy to improve performance as well as reduce complexity, but large patches may obscure the semantic information of time series and negatively impact the performance. Therefore, developing more efficient algorithms is especially crucial for facilitating large-scale time series analysis with LLMs and enhancing interactions with end users.

6.4 Combining Domain Knowledge

Combining existing statistical domain knowledge with LLMs may further boost the model’s capability for time series analysis. For example, TEMPO [Cao et al., 2023] applies time series seasonal-trend decomposition and treats decomposed components as different semantic inductive biases as input to the pre-trained transformer. FreqTST [Li et al., 2023] leverages insights from the frequency domain by tokenizing single time series into frequency units with weights for downstream forecasting. Further incorporating domain knowledge, such as wavelet decomposition, auto-correlation analysis, and empirical mode decomposition may augment LLMs’ capabilities in analyzing time series data.

6.5 Customization and Privacy

Existing works on large language models and time series analysis typically train a global model for all end users. Training customized models for different users based on the global model may bring further benefits and flexibility. Another important consideration is privacy, especially as many time series data are collected in private settings for clinical purposes or smart home applications. Federated learning offers a solution by enabling the training of machine learning models across multiple decentralized devices holding local data samples. Advancing research into model customization and user privacy preservation like federated learning would broaden the utility of LLM-empowered time series analysis.

7 Conclusion

We present the first survey that systematically analyzes the categorization of transferring knowledge from large language models for numerical time series analysis: direct prompting, time series quantization, aligning, the use of the vision modality to connect text and time series, and the integration of large language models with other analytical tools. For each category, we introduce their mathematical formulation, representative works, and compare their advantages and limitations. We also introduce representative multimodal text and time series datasets in various domains such as healthcare, IoT, finance, and audio. Concluding the paper, we outline the challenges and emerging directions for potential future research of LLM-empowered time series analysis.

Acknowledgements

Our work is supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Our work is also sponsored by NSF CAREER Award 2239440, NSF Prototype OKN Award 2333790, NIH Bridge2AI Center Program under award 1U54HG012510-01, Cisco-UCSD Sponsored Research Project, as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.
References


[Moon et al., 2023] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank...


