

Harnessing Vision Models for Time Series Analysis: A Survey

Jingchao Ni¹, Ziming Zhao¹, ChengAo Shen¹, Hanghang Tong², Dongjin Song³,
Wei Cheng⁴, Dongsheng Luo⁵, Haifeng Chen⁴

¹University of Houston, Houston, TX, USA

²University of Illinois at Urbana-Champaign, Urbana, IL, USA

³University of Connecticut, Storrs, CT, USA

⁴NEC Laboratories America, Princeton, NJ, USA

⁵Florida International University, Miami, FL, USA

{jni7, zzhao35, cshen9}@uh.edu, htong@illinois.edu,
dongjin.song@uconn.edu, {weicheng, haifeng}@nec-labs.com, dluo@fiu.edu

Abstract

Time series analysis has evolved from traditional autoregressive models to deep learning, Transformers, and Large Language Models (LLMs). While vision models have also been explored along the way, their contributions are less recognized due to the predominance of sequence modeling. However, challenges such as the mismatch between continuous time series and LLMs’ discrete token space, and the difficulty in capturing multivariate correlations, have led to growing interest in Large Vision Models (LVMs) and Vision-Language Models (VLMs). This survey highlights the advantages of vision models over LLMs in time series analysis, offering a comprehensive dual-view taxonomy that answers key research questions like how to encode time series as images and how to model imaged time series. Additionally, we address pre- and post-processing challenges in this framework and outline future directions for advancing the field.

1 Introduction

Vision models have long been applied to time series analysis. Early successes with 1D CNNs [Bai *et al.*, 2018], such as WaveNet [Van Den Oord *et al.*, 2016], led to their widespread use across time series tasks [Koprinska *et al.*, 2018; Zhang *et al.*, 2020]. Recently, advances in language-based sequence modeling have shifted attention to Transformers [Wen *et al.*, 2023] and Large Language Models (LLMs) [Zhang *et al.*, 2024]. Meanwhile, the demands for universal modeling has driven the emergence of time series foundation models such as TimesFM [Das *et al.*, 2024], Chronos [Ansari *et al.*, 2024], and Time-MoE [Shi *et al.*, 2024].

As Large Vision Models (LVMs), such as ViT [Dosovitskiy *et al.*, 2021], BEiT [Bao *et al.*, 2022] and MAE [He *et al.*, 2022], become achieving a similar success as LLMs (but in vision domain), a great deal of emergent efforts has been invested to explore the potential of LVMs in time series modeling [Chen *et al.*, 2024]. This is inspired by the plenty of ways for visualizing time series as images such as line plots

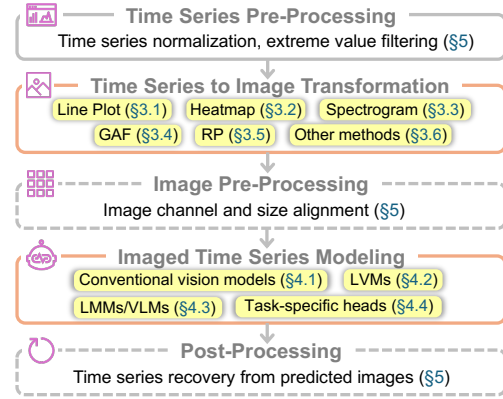


Figure 1: The general process of leveraging vision models for time series analysis. The red boxes are two views of taxonomy used in this survey. The dashed boxes denote optional, task-dependent steps.

of univariate time series (UTS) and heatmaps of multivariate time series (MTS). Such images provide a more straightforward view of time series than the counterpart textual representations to humans and, presumably, AI bots.

Taking a closer inspection reveals more advantages favoring LVMs over LLMs: (1) There is an inherent relationship between images and time series – each row/column in an image (per channel) is a sequence of *continuous* pixel values. By pre-training on massive images, LVMs may have learned important sequential patterns such as trends, periods, and spikes [Chen *et al.*, 2024]. In contrast, LLMs are pre-trained on *discrete* tokens, thus are less aligned with continuous time series. In fact, LLMs’ effectiveness on time series modeling is in question [Tan *et al.*, 2024]; (2) Instead of using channel-independence assumption [Nie *et al.*, 2023] to individually model each variate in an MTS, some imaging methods (§3.7) can naturally represent MTS, enabling explicit correlation encoding; (3) When prompting LLMs, existing methods often struggle with properly verbalizing a long sequence (or a matrix) of floating numbers in a UTS (or MTS), which may be limited by the context length or induce high API costs. In contrast, existing works find that using LVMs on imaged time

series is more prompt-friendly and less API-costly [Daswani *et al.*, 2024]; (4) Some imaging methods can encode long time series in a compact manner [Naiman *et al.*, 2024], thus have a great potential in modeling long-term dependency.

Also, the concurrent developments of LLMs and LVMs for time series pave the way for a confluence, *i.e.*, leveraging Large Multimodal Models (LMMs), such as LLaVA [Liu *et al.*, 2023], Gemini [Team, 2023] and Claude-3 [Anthropic, 2024], to consolidate the two complementary modalities, which may revolutionize the way (*e.g.*, visually, linguistically, *etc.*) that users interact with time series.

Despite the significance, a thorough review of relevant works is absent in the existing literature to the best of our knowledge. The survey [Zhang *et al.*, 2024] discusses a few vision models, but its focus is LLMs for time series. In light of this, in this survey, we comprehensively investigate the traditional and the state-of-the-art (SOTA) methods. Fig. 1 identifies the general process of applying vision models for time series analysis, which also serves as the structure of this survey. Our taxonomy has a dual view: (1) in Time Series to Image Transformation (§3), we review 5 primary *imaging methods* including Line Plot, Heatmap, Spectrogram, Gramian Angular Field (GAF), Recurrence Plot (RP), and some other methods; (2) in Imaged Time Series Modeling (§4), we discuss conventional vision models, LVMs and the initial efforts in LMMs. To highlight the taxonomy, we defer the discussion on the desiderata of pre- and post-processing to the end of this survey (§5). For comparison, we provide Table 1 to summarize the existing methods. Finally, we discuss future directions in this promising field (§6). A Github repository¹ is also maintained to provide up-to-date resources including our code of the imaging methods in §3. We hope this survey could be an orthogonal complement to the existing surveys on Transformer [Wen *et al.*, 2023], LLMs [Zhang *et al.*, 2024; Jiang *et al.*, 2024] and foundation models [Liang *et al.*, 2024] for time series, and provide a complete view on the process of using vision models for time series analysis, so as to be an insightful guidebook to the developers in this area.

2 Preliminaries and Taxonomy

In this paper, a UTS is represented by $\mathbf{x} = [x_1, \dots, x_T] \in \mathbb{R}^{1 \times T}$ where T is the length of the UTS, x_t ($1 \leq t \leq T$) is the value at time step t . Suppose there are d variates (or features), let $\mathbf{x}_i \in \mathbb{R}^{1 \times T}$ ($1 \leq i \leq d$) be a UTS of the i -th variate, an MTS can be represented by $\mathbf{X} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_d^\top]^\top \in \mathbb{R}^{d \times T}$.

As illustrated in Fig. 1, this survey focuses on methods that transform time series to images, namely *imaged time series*, and then apply vision models on the imaged time series for tackling time series tasks, such as classification, forecasting and anomaly detection. It is noteworthy that methods on videos or sequential images (*a.k.a.* image time series [Tarsiou *et al.*, 2023]) do not belong to this category because they don’t transform time series to images. Similarly, methods for spatiotemporal traffic data are out of our scope if the methods focus on streams of images (*e.g.*, traffic flows in a stream of grid maps [Zhang *et al.*, 2017]), but methods on imaging time-space matrices [Ma *et al.*, 2017] that resemble MTS are

included. For vision models on audios, this survey only discusses some representative works in §3.3 due to space limit. The focus of the survey will remain on general time series.

2.1 Taxonomy

We propose a taxonomy from the two views of *Time Series to Image Transformation* (§3) and *Imaged Time Series Modeling* (§4) as illustrated in Fig. 1. For the former, we discuss 5 primary methods for imaging UTS or MTS, and remark on their pros and cons. For the latter, we classify the existing methods by conventional vision models, LVMs and LMMs. We discuss their strategies on pre-training, fine-tuning, prompting, and the designs of task-specific heads. We also discuss the challenges and solutions in pre-/post-processing in §5. Table 1 presents a summary. In the following two sections, we will delve into the existing methods from the two views.

3 Time Series To Image Transformation

This section summarizes the methods for imaging time series (§3.1-§3.6) and their extensions to encode MTS (§3.7).

3.1 Line Plot

Line Plot is a straightforward way for visualizing UTS for human analysis (*e.g.*, stocks, power consumption, *etc.*). As illustrated by Fig. 2(a), the simplest approach is to draw a 2D image with x-axis representing time steps and y-axis representing time-wise values, with a line connecting all values of the series over time. This image can be either three-channel (*i.e.*, RGB) or single-channel as the colors may not be informative [Cohen *et al.*, 2020; Sood *et al.*, 2021; Zhang *et al.*, 2023]. For CNN [Semenoglou *et al.*, 2023] even uses a single 8-bit integer to represent each pixel for black-white images. So far, there is no consensus on whether other graphical components, such as legend, grids and tick labels, could provide extra benefits in any task. For example, ViTST [Li *et al.*, 2023b] finds these components are superfluous in a classification task, while TAMA [Zhuang *et al.*, 2024] finds grid-like auxiliary lines help enhance anomaly detection.

In addition to the regular Line Plot, MV-DTSA [Yang *et al.*, 2023] and ViTime [Yang *et al.*, 2024] divide an image into $h \times L$ grids, and define a function to map each time step of a UTS to a grid, producing a grid-like Line Plot. Also, we include methods that use Scatter Plot [Daswani *et al.*, 2024; Prithyani *et al.*, 2024] in this category because a Scatter Plot resembles a Line Plot but doesn’t connect data points with a line. By comparing them, [Prithyani *et al.*, 2024] finds a Line Plot could induce better time series classification.

For MTS, we defer the discussion on Line Plot to §3.7.

3.2 Heatmap

As shown in Fig. 2(b), a Heatmap visualizes the magnitudes of the values in matrix using color. It has been used to represent the matrix of MTS, *i.e.*, $\mathbf{X} \in \mathbb{R}^{d \times T}$, as a one-channel $d \times T$ image [Li *et al.*, 2022; Yazdanbakhsh and Dick, 2019]. TimEHR [Karami *et al.*, 2024] extends this to *irregular* MTS, where the intervals between time steps are uneven, by grouping the uneven time steps into H uniform bins, creating a $d \times H$ Heatmap image. In [Zeng *et al.*, 2021], a

¹<https://github.com/D2I-Group/awesome-vision-time-series>

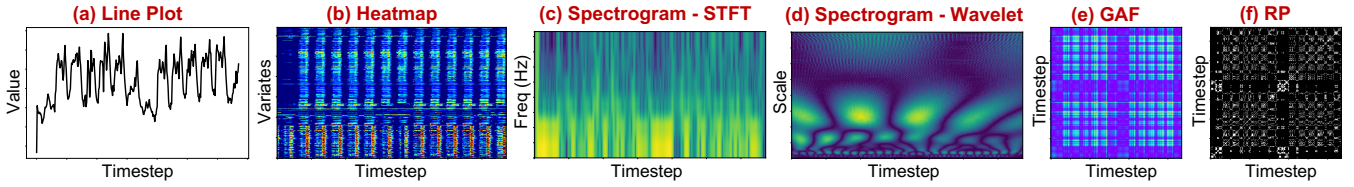


Figure 2: An illustration of different methods for imaging time series with a sample (length=336) from the *Electricity* benchmark dataset [Nie *et al.*, 2023]. (a)(c)(d)(e)(f) visualize the same variate. (b) visualizes all 321 variates. Filterbank is omitted due to its similarity to STFT.

different method is used for visualizing a 9-variate financial MTS. It reshapes the 9 variates at each time step to a 3×3 Heatmap, and uses a sequence of the images to forecast future frames, achieving time series forecasting. In contrast, VisionTS [Chen *et al.*, 2024] uses Heatmap to visualize UTS. Similar to TimesNet [Wu *et al.*, 2023] and TimeMixer++ [Wang *et al.*, 2025], it segments a length- T UTS into $\lfloor T/P \rfloor$ length- P subsequences, where P denotes a period, stacks them into a $P \times \lfloor T/P \rfloor$ matrix, and replicates the channel 3 times to form a grayscale image for input to an LVM. For MTS, VisionTS adopts channel-independence strategy [Nie *et al.*, 2023] and individually models each variate.

Remark. Heatmap can be used to visualize matrices of various forms. It is also used for matrices generated by the subsequent methods (e.g., Spectrogram, GAF, RP) in this section. In this paper, the name Heatmap refers specifically to images that use color to visualize the (normalized) values in UTS \mathbf{x} or MTS \mathbf{X} without performing other transformations.

3.3 Spectrogram

A *spectrogram* is a visual representation of the spectrum of frequencies of a signal as it varies with time, which are extensively used for analyzing audio signals [Gong *et al.*, 2021]. Since audio signals are a type of UTS, spectrogram can be considered as a method for imaging a UTS. As shown in Fig. 2(c), a common format is a 2D heatmap image with x-axis representing time steps and y-axis representing frequency, *a.k.a.* a time-frequency space. Each pixel in the image represents the (logarithmic) amplitude of a specific frequency at a specific time point. Typical methods for producing a spectrogram include **Short-Time Fourier Transform (STFT)** [Griffin and Lim, 1984], **Wavelet Transform** [Daubechies, 1990], and **Filterbank** [Vetterli and Herley, 1992].

STFT. Discrete Fourier transform (DFT) can be used to describe the intensity $f(w)$ of each constituent frequency w of a UTS signal $\mathbf{x} \in \mathbb{R}^{1 \times T}$. However, $f(w)$ has no time dependency. It cannot provide dynamic information such as when a specific frequency appears in the UTS. STFT addresses this deficiency by sliding a window function $g(t)$ over the time steps in \mathbf{x} , and computing the DFT within each window by

$$f(w, \tau) = \sum_{t=1}^T x_t g(t - \tau) e^{-iwt} \quad (1)$$

where w is frequency, τ is the position of the window, $f(w, \tau)$ describes the intensity of frequency w at time step τ .

By selecting a proper window function $g(\cdot)$ (e.g., Gaussian/Hamming/Bartlett window), a 2D spectrogram (e.g., Fig.

2(c)) can be drawn via a heatmap on the squared values $|f(w, \tau)|^2$, with w as the y-axis, and τ as the x-axis. For example, [Dixit *et al.*, 2024] uses STFT based spectrogram as an input to LMMs for time series classification.

Wavelet Transform. Continuous Wavelet Transform (CWT) uses the inner product to measure the similarity between a signal function $x(t)$ and an analyzing function. The analyzing function is a *wavelet* $\psi(t)$, where the typical choices include Morse wavelet, Morlet wavelet, *etc.* CWT compares $x(t)$ to the shifted and scaled (*i.e.*, stretched or shrunk) versions of the wavelet, and output a CWT coefficient by

$$c(s, \tau) = \int_{-\infty}^{\infty} x(t) \frac{1}{s} \psi^* \left(\frac{t - \tau}{s} \right) dt \quad (2)$$

where $*$ denotes complex conjugate, τ is the time step to shift, and s represents the scale. In practice, a discretized version of CWT in Eq. (2) is implemented for UTS $[x_1, \dots, x_T]$.

It is noteworthy that the scale s controls the frequency encoded in a wavelet – a larger s leads to a stretched wavelet with a lower frequency, and vice versa. As such, by varying s and τ , a 2D spectrogram (e.g., Fig. 2(d)) can be drawn on $|c(s, \tau)|$, where s is the y-axis and τ is the x-axis. Compared to STFT, which uses a fixed window size, Wavelet Transform allows variable wavelet sizes – a larger size for more precise low frequency information. Thus, the methods in [Du *et al.*, 2020; Namura *et al.*, 2024; Zeng *et al.*, 2023] choose CWT (with Morlet wavelet) to generate the spectrogram.

Filterbank. This method resembles STFT and is often used in processing audio signals. Given an audio signal, it firstly goes through a *pre-emphasis filter* to boost high frequencies, which helps improve the clarity of the signal. Then, STFT is applied on the signal. Finally, multiple “triangle-shaped” filters spaced on a Mel-scale are applied to the STFT power spectrum $|f(w, \tau)|^2$ to extract frequency bands. The outcome filterbank features $\hat{f}(w, \tau)$ can be used to yield a spectrogram with w as the y-axis, and τ as the x-axis.

Filterbank was adopted in AST [Gong *et al.*, 2021] with a 25ms Hamming window that shifts every 10ms for classifying audio signals using Vision Transformer (ViT). It then becomes widely used in the follow-up works such as SSAST [Gong *et al.*, 2022], MAE-AST [Baade *et al.*, 2022], and AST-SED [Li *et al.*, 2023a], as summarized in Table 1.

3.4 Gramian Angular Field (GAF)

GAF was introduced for classifying UTS using CNNs by [Wang and Oates, 2015a]. It was then extended to an imputation task in [Wang and Oates, 2015b]. Similarly, [Barra *et al.*, 2020] applied GAF for financial time series forecasting.

Method	TS-Type	Imaging	Imaged Time Series Modeling					TS-Recover	Task	Domain	Code
			Multimodal	Model	Pre-trained	Fine-tune	Prompt				
[Silva <i>et al.</i> , 2013]	UTS	RP	✗	K-NN	✗	✗	✗	✗	Classification	General	✗
[Wang and Oates, 2015a]	UTS	GAF	✗	CNN	✗	✓ ^b	✗	✓	Classification	General	✗
[Wang and Oates, 2015b]	UTS	GAF	✗	CNN	✗	✓ ^b	✗	✓	Multiple	General	✗
[Ma <i>et al.</i> , 2017]	MTS	Heatmap	✗	CNN	✗	✓ ^b	✗	✓	Forecasting	Traffic	✗
[Hatami <i>et al.</i> , 2018]	UTS	RP	✗	CNN	✗	✓ ^b	✗	✗	Classification	General	✗
[Yazdanbakhsh and Dick, 2019]	MTS	Heatmap	✗	CNN	✗	✓ ^b	✗	✗	Classification	General	✓ ^[1]
MSCRED [Zhang <i>et al.</i> , 2019]	MTS	Other (§3.6)	✗	ConvLSTM	✗	✓ ^b	✗	✗	Anomaly	General	✓ ^[2]
[Li <i>et al.</i> , 2020]	UTS	RP	✗	CNN	✓	✓	✗	✗	Forecasting	General	✓ ^[3]
[Cohen <i>et al.</i> , 2020]	UTS	LinePlot	✗	Ensemble	✗	✓ ^b	✗	✗	Classification	Finance	✗
[Barra <i>et al.</i> , 2020]	UTS	GAF	✗	CNN	✗	✓ ^b	✗	✗	Classification	Finance	✗
VisualAE [Sood <i>et al.</i> , 2021]	UTS	LinePlot	✗	CNN	✗	✓ ^b	✗	✓	Forecasting	Finance	✗
[Zeng <i>et al.</i> , 2021]	MTS	Heatmap	✗	CNN, LSTM	✗	✓ ^b	✗	✓	Forecasting	Finance	✗
AST [Gong <i>et al.</i> , 2021]	UTS	Spectrogram	✗	DeiT	✓	✓	✗	✗	Classification	Audio	✓ ^[4]
TTS-GAN [Li <i>et al.</i> , 2022]	MTS	Heatmap	✗	ViT	✗	✓ ^b	✗	✓	Ts-Generation	Health	✓ ^[5]
SSAST [Gong <i>et al.</i> , 2022]	UTS	Spectrogram	✗	ViT	✓ ^b	✓	✗	✗	Classification	Audio	✓ ^[6]
MAE-AST [Baade <i>et al.</i> , 2022]	UTS	Spectrogram	✗	MAE	✓ ^b	✓	✗	✗	Classification	Audio	✓ ^[7]
AST-SED [Li <i>et al.</i> , 2023a]	UTS	Spectrogram	✗	SSAST, GRU	✓	✓	✗	✗	EventDetection	Audio	✗
ForCNN [Semenoglou <i>et al.</i> , 2023]	UTS	LinePlot	✗	CNN	✗	✓ ^b	✗	✗	Forecasting	General	✗
Vit-num-spec [Zeng <i>et al.</i> , 2023]	UTS	Spectrogram	✗	ViT	✗	✓ ^b	✗	✗	Forecasting	Finance	✗
ViTST [Li <i>et al.</i> , 2023b]	MTS	LinePlot	✗	Swin	✓	✓	✗	✗	Classification	General	✓ ^[8]
MV-DTSA [Yang <i>et al.</i> , 2023]	UTS*	LinePlot	✗	CNN	✗	✓ ^b	✗	✓	Forecasting	General	✓ ^[9]
TimesNet [Wu <i>et al.</i> , 2023]	MTS	Heatmap	✗	CNN	✗	✓ ^b	✗	✓	Multiple	General	✓ ^[10]
ITF-TAD [Namura <i>et al.</i> , 2024]	UTS	Spectrogram	✗	CNN	✓	✗	✗	✗	Anomaly	General	✗
[Kaeprakmuk <i>et al.</i> , 2024]	UTS	GAF	✗	CNN	✓	✓	✗	✗	Classification	Sensing	✗
HCR-AdaAD [Lin <i>et al.</i> , 2024]	MTS	RP	✗	CNN, GNN	✗	✓ ^b	✗	✗	Anomaly	General	✗
FIRTS [Costa <i>et al.</i> , 2024]	UTS	Other (§3.6)	✗	CNN	✗	✓ ^b	✗	✗	Classification	General	✓ ^[11]
CAFO [Kim <i>et al.</i> , 2024]	MTS	RP	✗	CNN, ViT	✗	✓ ^b	✗	✗	Explanation	General	✓ ^[12]
ViTime [Yang <i>et al.</i> , 2024]	UTS*	LinePlot	✗	ViT	✓ ^b	✓	✗	✓	Forecasting	General	✓ ^[13]
ImagenTime [Naiman <i>et al.</i> , 2024]	MTS	Other (§3.6)	✗	CNN	✗	✓ ^b	✗	✓	Ts-Generation	General	✓ ^[14]
TimEHR [Karami <i>et al.</i> , 2024]	MTS	Heatmap	✗	CNN	✗	✓ ^b	✗	✓	Ts-Generation	Health	✓ ^[15]
VisionTS [Chen <i>et al.</i> , 2024]	UTS*	Heatmap	✗	MAE	✓	✓	✗	✓	Forecasting	General	✓ ^[16]
TimeMixer++ [Wang <i>et al.</i> , 2025]	MTS	Heatmap	✗	CNN	✗	✓ ^b	✗	✓	Multiple	General	✓ ^[17]
InsightMiner [Zhang <i>et al.</i> , 2023]	UTS	LinePlot	✓	LLaVA	✓	✓	✓	✗	Txt-Generation	General	✗
[Wimmer and Rekabsaz, 2023]	MTS	LinePlot	✓	CLIP, LSTM	✓	✓	✗	✗	Classification	Finance	✗
[Dixit <i>et al.</i> , 2024]	UTS	Spectrogram	✓	GPT4o, Gemini & Claude3	✓	✗	✓	✗	Classification	Audio	✗
[Daswani <i>et al.</i> , 2024]	MTS	LinePlot	✓	GPT4o, Gemini	✓	✗	✓	✗	Multiple	General	✗
TAMA [Zhuang <i>et al.</i> , 2024]	UTS	LinePlot	✓	GPT4o	✓	✗	✓	✗	Anomaly	General	✗
[Prithyani <i>et al.</i> , 2024]	MTS	LinePlot	✓	LLaVA	✓	✓	✓	✗	Classification	General	✓ ^[18]

Table 1: Taxonomy of vision models on time series. The top part includes unimodal models. The bottom part includes multimodal models. **TS-Type** denotes type of time series. **TS-Recover** denotes recovering time series from predicted images (§5). *: the method has been used to model the individual variates of an MTS. ^b: a new pre-trained model was proposed in the work. ^b: when pre-trained models were unused, “Fine-tune” refers to train a task-specific model from scratch. **Model** column: CNN could be regular CNN, ResNet, VGG-Net, *etc.*

Given a UTS $\mathbf{x} \in \mathbb{R}^{1 \times T}$, the first step is to rescale each x_t to a value \tilde{x}_t within $[0, 1]$ (or $[-1, 1]$). This range enables mapping \tilde{x}_t to polar coordinates by $\phi_t = \arccos(\tilde{x}_t)$, with a radius $r = t/N$ encoding the time stamp, where N is a constant factor to regularize the span of the polar coordinates. Two types of GAF, Gramian Sum Angular Field (GASF) and Gramian Difference Angular Field (GADF) are defined as

$$\begin{aligned} \text{GASF: } \cos(\phi_t + \phi_{t'}) &= x_t x_{t'} - \sqrt{1 - x_t^2} \sqrt{1 - x_{t'}^2} \\ \text{GADF: } \sin(\phi_t - \phi_{t'}) &= x_{t'} \sqrt{1 - x_t^2} - x_t \sqrt{1 - x_{t'}^2} \end{aligned} \quad (3)$$

which exploits the pairwise temporal correlations in the UTS. Thus, the outcome is a $T \times T$ matrix \mathbf{G} with $\mathbf{G}_{t,t'}$ specified by either type in Eq. (3). A GAF image is a heatmap on \mathbf{G} with both axes representing time, as illustrated by Fig. 2(e).

3.5 Recurrence Plot (RP)

RP [Eckmann *et al.*, 1987] encodes a UTS into an image that captures its periodic patterns by using its reconstructed *phase*

space. The phase space of $\mathbf{x} \in \mathbb{R}^{1 \times T}$ can be reconstructed by *time delay embedding* – a set of new vectors $\mathbf{v}_1, \dots, \mathbf{v}_l$ with

$$\mathbf{v}_t = [x_t, x_{t+\tau}, x_{t+2\tau}, \dots, x_{t+(m-1)\tau}] \in \mathbb{R}^{m\tau}, \quad 1 \leq t \leq l \quad (4)$$

where τ is the time delay, m is the dimension of the phase space, both are hyperparameters. Hence, $l = T - (m - 1)\tau$. With vectors $\mathbf{v}_1, \dots, \mathbf{v}_l$, an RP image measures their pairwise distances, results in an $l \times l$ image whose element is

$$\text{RP}_{i,j} = \Theta(\varepsilon - \|\mathbf{v}_i - \mathbf{v}_j\|), \quad 1 \leq i, j \leq l \quad (5)$$

where $\Theta(\cdot)$ is the Heaviside step function, ε is a threshold, and $\|\cdot\|$ is a norm function such as ℓ_2 norm. Eq. (5) generates a binary matrix with $\text{RP}_{i,j} = 1$ if \mathbf{v}_i and \mathbf{v}_j are sufficiently similar, producing a black-white image (*e.g.*, Fig. 2(f)).

An advantage of RP is its flexibility in image size by tuning m and τ . Thus it has been used for time series classification [Silva *et al.*, 2013; Hatami *et al.*, 2018], forecasting [Li *et al.*, 2020], anomaly detection [Lin *et al.*, 2024] and explanation [Kim *et al.*, 2024]. The method in [Hatami *et al.*, 2018], and

Method	TS-Type	Advantages	Limitations
Line Plot (§3.1)	UTS, MTS	matches human perception of time series	limited to MTS with a small number of variates
Heatmap (§3.2)	UTS, MTS	straightforward for both UTS and MTS	the order of variates may affect their correlation learning
Spectrogram (§3.3)	UTS	encodes the time-frequency space	limited to UTS; needs a proper choice of window/wavelet
GAF (§3.4)	UTS	encodes the temporal correlations in a UTS	limited to UTS; $O(T^2)$ time and space complexity
RP (§3.5)	UTS	flexibility in image size by tuning m and τ	limited to UTS; information loss after thresholding

Table 2: Summary of the five primary methods for transforming time series to images. **TS-Type** denotes type of time series.

similarly in HCR-AdaAD [Lin *et al.*, 2024], omit the thresholding in Eq. (5) and uses $\|\mathbf{v}_i - \mathbf{v}_j\|$ to produce continuously valued images to avoid information loss.

3.6 Other Methods

Additionally, [Wang and Oates, 2015a] introduces Markov Transition Field (MTF). MTF is a matrix $\mathbf{M} \in \mathbb{R}^{Q \times Q}$ encoding transition probabilities among Q time segments, to visualize UTS. ImagenTime [Naiman *et al.*, 2024] stacks the delay embeddings $\mathbf{v}_1, \dots, \mathbf{v}_l$ in Eq. (4) to an $l \times m\tau$ matrix for visualizing UTS. MSCRED [Zhang *et al.*, 2019] applies heatmaps to $d \times d$ correlation matrices of d -variate MTS for anomaly detection. Earlier, [Kumar *et al.*, 2005] proposes Bitmaps to image discretized UTS and defines distances among them. Some methods use a mixture of imaging methods for richer representations, *e.g.*, [Wang and Oates, 2015b] stacks GASF, GADF, and MTF into a 3-channel image. FIRTIS [Costa *et al.*, 2024] combines GASF, MTF, and RP. These multi-view representations have shown greater robustness than single-view images in these works for classification tasks.

3.7 How to Model MTS

In the above methods, Heatmap (§3.2) can be used to visualize the variate-time matrices, \mathbf{X} , of MTS (*e.g.*, Fig. 1(b)), where correlated variates should be spatially close to each other. Line Plot (§3.1) can be used to visualize MTS by plotting all variates in the same image [Wimmer and Rekabsaz, 2023; Daswani *et al.*, 2024] or combining all univariate images to compose a bigger image [Li *et al.*, 2023b], but these methods only work for a small number of variates. Spectrogram (§3.3), GAF (§3.4), and RP (§3.5) were designed specifically for UTS. For these methods and Line Plot, which are not straightforward in imaging MTS, the general approaches include using channel independence assumption to model each variate individually [Nie *et al.*, 2023], or stacking the images of d variates to form a d -channel image [Naiman *et al.*, 2024; Kim *et al.*, 2024]. However, the latter does not fit some vision models pre-trained on RGB images which requires 3-channel inputs (more discussions are deferred to §5).

Remark. As a summary, Table 2 recaps the salient advantages and limitations of the five primary imaging methods that are introduced in this section.

4 Imaged Time Series Modeling

With image representations, time series analysis can be readily performed with vision models. This section discusses such solutions from traditional models to the SOTA models.

4.1 Conventional Vision Models

Following traditional image classification, [Silva *et al.*, 2013] applies a K-NN classifier on the RPs of time series, [Cohen *et al.*, 2020] applies an ensemble of fundamental classifiers such as SVM and AdaBoost on the Line Plots for time series classification. As an image encoder, CNNs have been widely used for learning image representations. Different from using 1D CNNs on sequences [Bai *et al.*, 2018], 2D and 3D CNNs can be applied on imaged time series as shown in Fig. 3(a). For example, regular CNNs have been used on Spectrograms [Du *et al.*, 2020] and Heatmaps [Wang *et al.*, 2025], tiled CNNs have been used on GAF images [Wang and Oates, 2015a; Wang and Oates, 2015b], dilated CNNs have been used on Heatmaps [Yazdanbakhsh and Dick, 2019]. More frequently, ResNet [He *et al.*, 2016], Inception-v1 [Szegedy *et al.*, 2015], and VGG-Net [Simonyan and Zisserman, 2015] have been used on Line Plots [Semenoglou *et al.*, 2023], Heatmap images [Zeng *et al.*, 2021; Wu *et al.*, 2023], RP images [Li *et al.*, 2020; Kim *et al.*, 2024], GAF images [Barra *et al.*, 2020; Kaewrakmuk *et al.*, 2024], and even a mixture of GAF, MTF and RP images [Costa *et al.*, 2024]. In particular, for time series generation tasks, GAN frameworks of CNNs [Li *et al.*, 2022; Karami *et al.*, 2024] and a diffusion model with U-Nets [Naiman *et al.*, 2024] have also been explored.

Due to their small to medium sizes, these models are often trained from scratch using task-specific training data. Meanwhile, fine-tuning *pre-trained vision models* have already been found promising in cross-modal knowledge transfer for time series anomaly detection [Namura *et al.*, 2024] and forecasting [Li *et al.*, 2020].

4.2 Large Vision Models (LVMs)

Vision Transformer (ViT) [Dosovitskiy *et al.*, 2021] has inspired the development of modern LVMs such as Swin [Liu *et al.*, 2021], BEiT [Bao *et al.*, 2022], and MAE [He *et al.*, 2022]. As Fig. 3(b) shows, ViT splits an image into *patches* of fixed size, then embeds each patch and augments it with a positional embedding. The vectors of patches are processed by a Transformer as if they were token embeddings. Compared to CNNs, ViTs are less data-efficient, but have higher capacity. Thus, *pre-trained ViTs* have been explored for modeling imaged time series. For example, AST [Gong *et al.*, 2021] fine-tunes DeiT [Touvron *et al.*, 2021] on the filterbank spectrogram of audios for classification tasks and finds ImageNet-pretrained DeiT is remarkably effective in knowledge transfer. The fine-tuning paradigm has also been adopted in [Zeng *et al.*, 2023; Li *et al.*, 2023b] but with different pre-trained models such as Swin by [Li *et al.*, 2023b]. VisionTS [Chen *et al.*, 2024] attributes LVMs’ superiority over

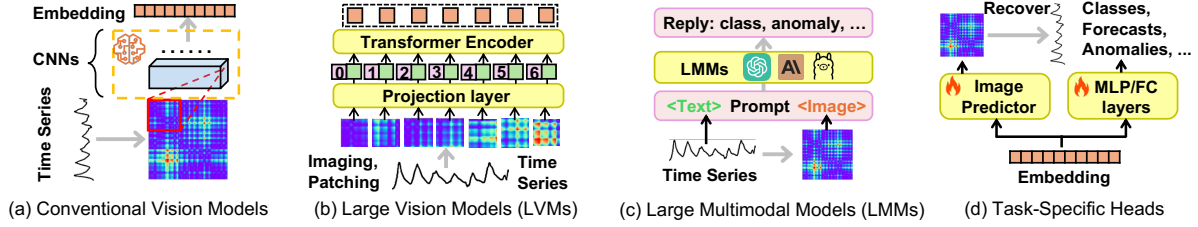


Figure 3: An illustration of different modeling strategies on imaged time series in (a)(b)(c) and task-specific heads in (d).

LLMs in knowledge transfer to the small gap between the pre-trained images and imaged time series. It finds that with one-epoch fine-tuning, MAE becomes the SOTA time series forecasters on some benchmark datasets.

Similar to time series foundation models such as TimesFM [Das *et al.*, 2024], there are some initial efforts in pre-training ViT architectures with imaged time series. Following AST, SSAST [Gong *et al.*, 2022] introduced a masked spectrogram patch prediction framework for pre-training ViT on a large dataset – AudioSet-2M. Then it becomes a backbone of some follow-up works such as AST-SED [Li *et al.*, 2023a] for sound event detection. For UTS, ViTime [Yang *et al.*, 2024] generates a large set of Line Plots of synthetic UTS for pre-training ViT, which was found superior over TimesFM in zero-shot forecasting tasks on benchmark datasets.

4.3 Large Multimodal Models (LMMs)

As LMMs get growing attentions, some notable LMMs, such as LLaVA [Liu *et al.*, 2023], Gemini [Team, 2023], GPT-4o [Achiam *et al.*, 2023] and Claude-3 [Anthropic, 2024], have been explored to consolidate the power of LLMs and LVMs for time series analysis. Since LMMs support multimodal input via prompts, methods in this thread typically prompt LMMs with textual and imaged representations of time series, and instructions on what tasks to perform (*e.g.*, Fig. 3(c)).

InsightMiner [Zhang *et al.*, 2023] is a pioneer work that uses the LLaVA architecture to generate texts describing the trend of each input UTS. It extracts the trend of a UTS by Seasonal-Trend decomposition, encodes the Line Plot of the trend, and concatenates the embedding of the Line Plot with the embeddings of a textual instruction, which includes a sequence of numbers representing the UTS, *e.g.*, “[1.1, 1.7, ..., 0.3]”. The concatenated embeddings are taken by a language model for generating trend descriptions. Similarly, [Prithyani *et al.*, 2024] adopts the LLaVA architecture, but for MTS classification. An MTS is encoded by the visual embeddings of the stacked Line Plots of all variates. The matrix of the MTS is also verbalized in a prompt as the textual modality. By integrating token embeddings, both methods fine-tune some layers of the LMMs with some synthetic data.

Moreover, zero-shot and in-context learning performance of several commercial LMMs have been evaluated for audio classification [Dixit *et al.*, 2024], anomaly detection [Zhuang *et al.*, 2024], and some synthetic tasks [Daswani *et al.*, 2024], where the image and textual representations of a query time series are integrated into a prompt. For in-context learning, these methods inject the images of a few example time series and their labels (*e.g.*, classes) into an instruction to prompt

LMMs for assisting the prediction of the query time series.

4.4 Task-Specific Heads

As Fig. 3(d) illustrates, for classification tasks, most of the methods in Table 1 adopt a fully connected (FC) layer or multilayer perceptron (MLP) to transform an embedding into a probability distribution over all classes. For forecasting tasks, there are two approaches: (1) using a $d_e \times W$ MLP/FC layer to directly predict (from the d_e -dimensional embedding) the time series values in a future time window of size W [Li *et al.*, 2020; Semoglou *et al.*, 2023]; (2) predicting the pixel values that represent the future part of the time series and then recovering the time series from the predicted image [Yang *et al.*, 2023; Chen *et al.*, 2024; Yang *et al.*, 2024] (§5 discusses the recovery methods). Imputation and generation tasks resemble forecasting as they also predict time series values. Thus approach (2) has been used for imputation [Wang and Oates, 2015b] and generation [Naiman *et al.*, 2024; Karami *et al.*, 2024]. When using LMMs for classification, text generation, and anomaly detection, most of the methods prompt LMMs to produce the desired outputs in textual answers, circumventing task-specific heads [Zhang *et al.*, 2023; Dixit *et al.*, 2024; Zhuang *et al.*, 2024].

5 Pre-Processing and Post-Processing

To be successful in using vision models, some subtle design desiderata include **time series normalization**, **image alignment** and **time series recovery**.

Time Series Normalization. Vision models are usually trained on standardized images. To be aligned, the images introduced in §3 should be normalized with a controlled mean and standard deviation, as did by [Gong *et al.*, 2021] on spectrograms. In particular, as Heatmap is built on raw time series values, the commonly used Instance Normalization (IN) [Kim *et al.*, 2022] can be applied on the time series as suggested by VisionTS [Chen *et al.*, 2024] since IN share similar merits as Standardization. Using Line Plot requires a proper range of y-axis. In addition to rescaling time series [Zhuang *et al.*, 2024], ViTST [Li *et al.*, 2023b] introduced several methods to remove extreme values from the plot. GAF requires min-max normalization on its input, as it transforms time series values within $[0, 1]$ to polar coordinates (*i.e.*, arcs-cos). In contrast, input to RP is usually normalization-free as an ℓ_2 norm is involved in Eq. (5) before thresholding.

Image Alignment. When using pre-trained models, it is imperative to fit the image size to the input requirement of the models. This is especially true for Transformer based models

as they use a fixed number of positional embeddings to encode the spatial information of image patches. For 3-channel RGB images such as Line Plot, it is straightforward to meet a pre-defined size by adjusting the resolution when producing the image. For images built upon matrices such as Heatmap, Spectrogram, GAF, RP, the number of channels and matrix size need adjustment. For the channels, one method is to duplicate a matrix to 3 channels [Chen *et al.*, 2024], another way is to average the weights of the 3-channel patch embedding layer into a 1-channel layer [Gong *et al.*, 2021]. For the image size, bilinear interpolation is a common method to resize input images [Chen *et al.*, 2024]. Alternatively, AST [Gong *et al.*, 2021] resizes the positional embeddings instead of the images to fit the model to a desired input size. However, the interpolation in these methods may either alter the time series or the spatial information in positional embeddings.

Time Series Recovery. As stated in §4.4, tasks such as forecasting, imputation and generation requires predicting time series values. For models that predict pixel values of images, post-processing involves recovering time series from the predicted images. Recovery from Line Plots is tricky, it requires locating pixels that represent time series and mapping them back to the original values. This can be done by manipulating a grid-like Line Plot as introduced in [Yang *et al.*, 2023; Yang *et al.*, 2024], which has a recovery function. In contrast, recovery from Heatmap is straightforward as it directly stores the predicted time series values [Zeng *et al.*, 2021; Chen *et al.*, 2024]. Spectrogram is underexplored in these tasks and it remains open on how to recover time series from it. The existing work [Zeng *et al.*, 2023] uses Spectrogram for forecasting only with an MLP head that directly predicts time series. GAF supports accurate recovery by an inverse mapping from polar coordinates to normalized time series [Wang and Oates, 2015b]. However, RP lost time series information during thresholding (Eq. 5), thus may not fit recovery-demanded tasks without using an *ad-hoc* prediction head.

6 Challenges and Future Directions

Fundamental Understanding. With various imaging methods available, most existing works select them based on intuition. There remains a gap in both theoretical and empirical understanding of questions such as which imaging methods fit which tasks and whether LVMs truly learn patterns that make them better suited to time series than LLMs. Some existing works evaluate multiple imaging methods, but in limited tasks. For example, ImagenTime [Naiman *et al.*, 2024] compares the representation abilities of GAF, STFT, and delay embedding (§3.6) in a time series generation task. However, a thorough understanding to guide the development of LVMs and LMMs across imaging methods is absent. This survey provides an initial comparative discussion in §3. Further empirical and theoretical investigations are essential to the synergy between LVMs/LMMs and time series analysis.

Modeling the Correlation of Variates in MTS. §3.7 reviewed existing methods for imaging MTS, each with limitations. For example, Heatmaps (*e.g.*, Fig. 2(b)) encode spatial relationships, so the row order of variates affects how correlations are modeled, implying correlated variates should

be spatially close to each other. Similarly, Line Plots do not explicitly capture inter-variate correlations. Stacking one channel per variate into a d -channel input hinders the use of pre-trained LVMs, which expect 3-channel RGB inputs. Therefore, there is a need for more effective imaging or modeling techniques (*e.g.*, incorporating graph neural networks (GNNs) on variates) to better capture correlations in MTS.

Advanced Imaging for Time Series. In addition to the basic methods introduced in §3, advanced image representations hold promise. For example, InsightMiner [Zhang *et al.*, 2023] adopts seasonal-trend decomposition, which is often used to extract components that can serve as inductive biases for time series models. Extending this idea to decompose images such as Spectrogram, GAF, and RP into finer-grained components may enhance vision models’ effectiveness. Moreover, combining multiple imaging methods from different views, such as frequency (Spectrogram), temporal structure (GAF), and recurrence patterns (RP), can provide richer representations. FIRTS [Costa *et al.*, 2024] stacks images across channels for a classification task, but is limited to images of the same size. Modeling a mixture of arbitrary images by methods such as multi-view learning may enable more flexibility.

Multimodal Time Series Models and Agents. As can be seen from Table 1, the existing research on multimodal analysis (with vision modality) is much less than unimodal analysis, with a limited scope of time series tasks. Given the existing LLMs for time series such as Time-LLM [Jin *et al.*, 2024] and S²IP [Pan *et al.*, 2024], it is appealing to introduce vision modality to further boost the performance in wide tasks such as forecasting, classification and anomaly detection. Furthermore, the visual representation of time series provides the foundation for exploring multimodal AI agents [Xie *et al.*, 2024] for more intricate and nuanced tasks that requires reasoning and interactions with environments, such as root cause analysis in AI for IT Operations (AIOps).

Vision-based Time Series Foundation Models. A foundation model (FM) is a deep learning model trained on vast datasets that is applicable to a wide range of tasks. Recent time series FMs, such as TimesFM [Das *et al.*, 2024], MOMENT [Goswami *et al.*, 2024], Chronos [Ansari *et al.*, 2024] and Time-MoE [Shi *et al.*, 2024], are mostly built upon LLM architectures and trained on raw time series. Given the potential of image representation, it is promising to explore vision models as a new architecture to revolutionize time series FMs. This research direction not only leverages the advantages of LVMs as introduced in §1 (*e.g.*, the prior knowledge extracted from the vast pre-training images), but also enables future development of vision-language FMs for time series.

7 Conclusion

In this paper, we present the first survey on leveraging vision models for time series analysis, whose general process structures the survey. We propose a new taxonomy consisting of imaging and modeling methods for time series. We discuss the pre- and post-processing steps as well. Each category encompasses representative methods and relevant remarks. The survey also highlights the challenges and future directions for further advancing time series analysis with vision models.

Acknowledgments

This work was partially supported by NSF (2134079), CA-REER Award IIS-2338878, IIS-2331908, and research gifts from NEC Laboratories America and Morgan Stanley.

References

- [Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, et al. Gpt-4 technical report. *arXiv:2303.08774*, 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.
- [Anthropic, 2024] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.
- [Baade *et al.*, 2022] Alan Baade, Puyuan Peng, and David Harwath. Mae-ast: Masked autoencoding audio spectrogram transformer. *arXiv:2203.16691*, 2022.
- [Bai *et al.*, 2018] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.
- [Bao *et al.*, 2022] Hangbo Bao, Li Dong, Songhao Piao, et al. Beit: Bert pre-training of image transformers. In *ICLR*, 2022.
- [Barra *et al.*, 2020] Silvio Barra, Salvatore Mario Carta, Andrea Corriga, Alessandro Sebastian Podda, et al. Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA J. Autom. Sin.*, 7(3):683–692, 2020.
- [Chen *et al.*, 2024] Mouxiang Chen, Lefei Shen, Zhuo Li, et al. Visions: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv:2408.17253*, 2024.
- [Cohen *et al.*, 2020] Naftali Cohen, Tucker Balch, and Manuela Veloso. Trading via image classification. In *ICAIF*, 2020.
- [Costa *et al.*, 2024] Henrique V Costa, André GR Ribeiro, and Vinicius MA Souza. Fusion of image representations for time series classification with deep learning. In *ICANN*, 2024.
- [Das *et al.*, 2024] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *ICML*, 2024.
- [Daswani *et al.*, 2024] Mayank Daswani, Mathias MJ Bellaiche, Marc Wilson, Desislav Ivanov, et al. Plots unlock time-series understanding in multimodal models. *arXiv:2410.02637*, 2024.
- [Daubechies, 1990] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inf. Theory*, 36(5):961–1005, 1990.
- [Dixit *et al.*, 2024] Satvik Dixit, Laurie Heller, and Chris Donahue. Vision language models are few-shot audio spectrogram classifiers. In *NeurIPS Workshop*, 2024.
- [Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [Du *et al.*, 2020] Bairui Du, Delmiro Fernandez-Reyes, and Paolo Barucca. Image processing tools for financial time series classification. *arXiv:2008.06042*, 2020.
- [Eckmann *et al.*, 1987] J-P Eckmann, S Oliffson Kamphorst, et al. Recurrence plots of dynamical systems. *EPL*, 4(9):973, 1987.
- [Gong *et al.*, 2021] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *Interspeech*, 2021.
- [Gong *et al.*, 2022] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Sast: Self-supervised audio spectrogram transformer. In *AAAI*, 2022.
- [Goswami *et al.*, 2024] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *ICML*, 2024.
- [Griffin and Lim, 1984] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Trans. Acoust.*, 32(2):236–243, 1984.
- [Hatami *et al.*, 2018] Nima Hatami, Yann Gavet, and Johan De-bayle. Classification of time-series images using deep convolutional neural networks. In *ICMV*, 2018.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. Deep residual learning for image recognition. In *CVPR*, 2016.
- [He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.
- [Jiang *et al.*, 2024] Yushan Jiang, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, et al. Empowering time series analysis with large language models: A survey. In *IJCAI*, 2024.
- [Jin *et al.*, 2024] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, et al. Time-llm: Time series forecasting by re-programming large language models. In *ICLR*, 2024.
- [Kaewrakmuk *et al.*, 2024] Thossapon Kaewrakmuk, Jakkree Sri-nonchat, et al. Multi-sensor data fusion and time series to image encoding for hardness recognition. *IEEE Sens. J.*, 2024.
- [Karami *et al.*, 2024] Hojjat Karami, Mary-Anne Hartley, David Atienza, et al. Timehr: Image-based time series generation for electronic health records. *arXiv:2402.06318*, 2024.
- [Kim *et al.*, 2022] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheon-bok Park, et al. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2022.
- [Kim *et al.*, 2024] Jaeho Kim, Seok-Ju Hahn, Yoontae Hwang, Junghye Lee, and Seulki Lee. Cafo: Feature-centric explanation on time series classification. In *KDD*, 2024.
- [Koprinska *et al.*, 2018] Irena Koprinska, Dongsong Wu, and Zheng Wang. Convolutional neural networks for energy time series forecasting. In *IJCNN*, 2018.
- [Kumar *et al.*, 2005] Nitin Kumar, Venkata Nishanth Lolla, Eamonn Keogh, et al. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SDM*, 2005.
- [Li *et al.*, 2020] Xixi Li, Yanfei Kang, and Feng Li. Forecasting with time series imaging. *Expert Syst. Appl.*, 160:113680, 2020.
- [Li *et al.*, 2022] Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. Tts-gan: A transformer-based time-series generative adversarial network. In *AIME*, 2022.
- [Li *et al.*, 2023a] Kang Li, Yan Song, Li-Rong Dai, et al. Ast-sed: An effective sound event detection method based on audio spectrogram transformer. In *ICASSP*, 2023.
- [Li *et al.*, 2023b] Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. *NeurIPS*, 2023.
- [Liang *et al.*, 2024] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, et al. Foundation models for time series analysis: A tutorial and survey. In *KDD*, 2024.
- [Lin *et al.*, 2024] Chunming Lin, Bowen Du, Leilei Sun, and Linchao Li. Hierarchical context representation and self-adaptive thresholding for multivariate anomaly detection. *TKDE*, 2024.

- [Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.
- [Liu *et al.*, 2023] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.
- [Ma *et al.*, 2017] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.
- [Naiman *et al.*, 2024] Ilan Naiman, Nimrod Berman, Itai Pemper, Idan Arbiv, Gal Fadlon, and Omri Azencot. Utilizing image transforms and diffusion models for generative modeling of short and long time series. *NeurIPS*, 2024.
- [Namura *et al.*, 2024] Nobuo Namura, Yuma Ichikawa, et al. Training-free time-series anomaly detection: Leveraging image foundation models. *arXiv preprint arXiv:2408.14756*, 2024.
- [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, et al. S²ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *ICML*, 2024.
- [Prithyani *et al.*, 2024] Vinay Prithyani, Mohsin Mohammed, Richa Gadgil, et al. On the feasibility of vision-language models for time-series classification. *arXiv:2412.17304*, 2024.
- [Semenoglou *et al.*, 2023] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. Image-based time series forecasting: A deep convolutional neural network approach. *Neural Netw.*, 157:39–53, 2023.
- [Shi *et al.*, 2024] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, et al. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv:2409.16040*, 2024.
- [Silva *et al.*, 2013] Diego F Silva, Vinícius MA De Souza, and Gustavo EAPA Batista. Time series classification using compression distance of recurrence plots. In *ICDM*, 2013.
- [Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [Sood *et al.*, 2021] Srijan Sood, Zhen Zeng, Naftali Cohen, Tucker Balch, and Manuela Veloso. Visual time series forecasting: an image-driven approach. In *ICAIF*, 2021.
- [Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, et al. Going deeper with convolutions. In *CVPR*, 2015.
- [Tan *et al.*, 2024] Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *NeurIPS*, 2024.
- [Tarasiou *et al.*, 2023] Michail Tarasiou, Erik Chavez, and Stefanos Zafeiriou. Vits for sits: Vision transformers for satellite image time series. In *CVPR*, 2023.
- [Team, 2023] Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.
- [Touvron *et al.*, 2021] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, et al. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.
- [Van Den Oord *et al.*, 2016] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, et al. Wavenet: A generative model for raw audio. *arXiv:1609.03499*, 12, 2016.
- [Vetterli and Herley, 1992] Martin Vetterli and Cormac Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. Signal Process.*, 40(9):2207–2232, 1992.
- [Wang and Oates, 2015a] Zhiguang Wang and Tim Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *AAAI Workshop*, 2015.
- [Wang and Oates, 2015b] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In *IJCAI*, 2015.
- [Wang *et al.*, 2025] Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenzhe Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *ICLR*, 2025.
- [Wen *et al.*, 2023] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: a survey. In *IJCAI*, 2023.
- [Wimmer and Rekabsaz, 2023] Christopher Wimmer and Navid Rekabsaz. Leveraging vision-language models for granular market change prediction. *arXiv:2301.10166*, 2023.
- [Wu *et al.*, 2023] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, et al. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.
- [Xie *et al.*, 2024] Junlin Xie, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. Large multimodal agents: A survey. *arXiv:2402.15116*, 2024.
- [Yang *et al.*, 2023] Luoxiao Yang, Xinqi Fan, et al. Your time series is worth a binary image: machine vision assisted deep framework for time series forecasting. *arXiv:2302.14390*, 2023.
- [Yang *et al.*, 2024] Luoxiao Yang, Yun Wang, Xinqi Fan, Israel Cohen, et al. Vitime: A visual intelligence-based foundation model for time series forecasting. *arXiv:2407.07311*, 2024.
- [Yazdanbakhsh and Dick, 2019] Omolbanin Yazdanbakhsh and Scott Dick. Multivariate time series classification using dilated convolutional neural network. *arXiv:1905.01697*, 2019.
- [Zeng *et al.*, 2021] Zhen Zeng, Tucker Balch, et al. Deep video prediction for time series forecasting. In *ICAIF*, 2021.
- [Zeng *et al.*, 2023] Zhen Zeng, Rachneet Kaur, Suchetha Siddagangappa, et al. From pixels to predictions: Spectrogram and vision transformer for better time series forecasting. In *ICAIF*, 2023.
- [Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, 2017.
- [Zhang *et al.*, 2019] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI*, 2019.
- [Zhang *et al.*, 2020] Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *AAAI*, 2020.
- [Zhang *et al.*, 2023] Yunkai Zhang, Yawen Zhang, et al. Insight miner: A time series analysis dataset for cross-domain alignment with natural language. In *NeurIPS AI4Science*, 2023.
- [Zhang *et al.*, 2024] Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: A survey. In *IJCAI*, 2024.
- [Zhuang *et al.*, 2024] Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, et al. See it, think it, sorted: Large multimodal models are few-shot time series anomaly analyzers. *arXiv:2411.02465*, 2024.