Artificial Intelligence Review (2025) 58:80
https://doi.org/10.1007/s10462-024-11044-2

A systematic review for transformer-based long-term series forecasting

Liyilei Su[1,2] · Xumin Zuo[1,2] · Rui Li[1] · Xin Wang[1] · Heng Zhao[1] · Bingding Huang[1,2]

Accepted: 29 November 2024 / Published online: 6 January 2025

Abstract
The emergence of deep learning has yielded noteworthy advancements in time series forecasting (TSF). Transformer architectures have witnessed broad utilization and adoption in TSF tasks. Transformers have proven to be the most successful solution to extract the semantic correlations among the elements within a long sequence. Various variants have enabled Transformer architecture to effectively handle long-term time series forecasting (LTSF) tasks. In this article, we first present a comprehensive overview of Transformer architectures and their subsequent enhancements developed to address various LTSF tasks. Then, we summarize the publicly available LTSF datasets and relevant evaluation metrics. Furthermore, we provide valuable insights into the best practices and techniques for effectively training Transformers in the context of time-series analysis. Lastly, we propose potential research directions in this rapidly evolving field.

Keywords Long-term time series forecasting · Deep learning · Transformer · Self-attention · Multi-head attention

1 Introduction

The time series is usually a set of random variables observed and recorded sequentially over time. Key research directions for time-series data are classification [1, 2], anomaly detection [3–5], event prediction [6–8], and time series forecasting [9–11]. Time series forecasting (TSF) predicts the future trend changes of time series from a large amount of data in various fields. With the development of data collection technology, the task gradually evolves into

Liyilei Su and Xumin Zuo have been contributed equally to this work.

✉ Heng Zhao
zhaoheng@sztu.edu.cn

✉ Bingding Huang
huangbingding@sztu.edu.cn

[1] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518188, China

[2] College of Applied Sciences, Shenzhen University, Shenzhen 518060, China

using more historical data to predict the longer-term future, which is long-term time series forecasting (LTSF) [12, 13]. Precise LTSF can offer support to decision makers to better plan for the future by forecasting outcomes further in advance, including meteorology prediction [14], noise cancellation [15], financial long-term strategic guidance [16], power load forecasting [17, 18], and traffic road condition prediction [19].

Formerly, traditional statistical approaches were applied to time series forecasting, such as autoregressive (AR) [20], moving average (MA) [21] models, auto-regressive moving average (ARMA) [22], AR Integrated MA (ARIMA) [23], and spectral analysis techniques [24]. However, these traditional statistical methods require many a priori assumptions on the time-series prediction, such as stability, normal distribution, linear correlation, and independence. For example, AR, MA, and ARMA models are based on the assumption that time series are stationary, but in many real cases, time-series data exhibit non-stationarity. These assumptions limit the effectiveness of these traditional methods in real-world applications.

As it is difficult to effectively capture the nonlinear relationships between time series with traditional statistical approaches, many researchers have studied LTSF from the perspective of machine learning (ML) [25–29]. Support vector machines (SVMs) [30] and adaptive boosting (AdaBoost) [31] were employed in the field of TSF. They calculate data metrics, such as minimum, maximum, mean, and variance, within a sliding window as new features for prediction. These models have somewhat solved the problem of predicting multivariate, heteroskedastic time series with nonlinear relationships. However, they suffer from poor generalization, which leads to limited prediction accuracy.

Deep learning (DL) models (Fig. 1) have greatly improved the nonlinear modeling capabilities of TSF in recent years. These models are constructed with neural network structures with powerful nonlinear modeling capabilities to learn complex patterns and feature representations in time series automatically. Therefore, DL is an effective solution for TSF and many other problems related to TSF, such as hierarchical time series forecasting [32], intermittent time series forecasting [33], and sparse multivariate time series forecasting [34] asynchronous time series forecasting [35, 36]. It has extended some multi-objective, multi-granular forecasting scenarios [37] and multi-modal time series forecasting scenarios [38,

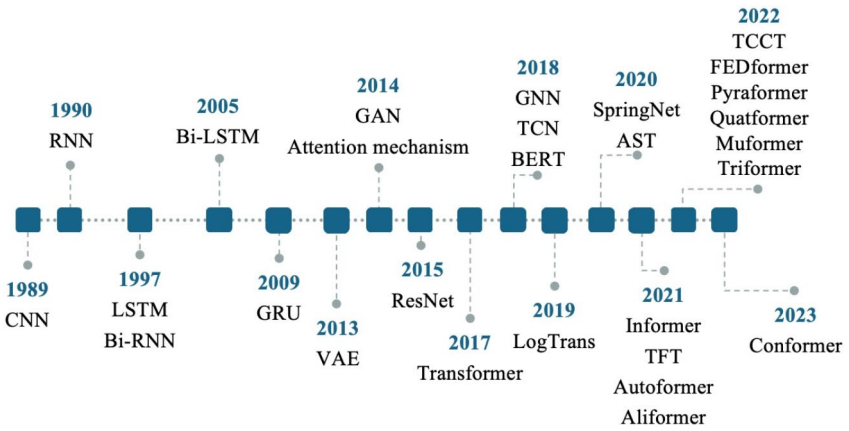

Fig. 1 The development history of TSF algorithms based on deep learning

39]. The advantage of deep learning models can be attributed to their profound flexibility and ability to capture long-term dependencies and handle large-scale data.

It is noteworthy that recurrent neural networks (RNNs) [40] and their variants, such as long short-term memory networks (LSTMs) [41] and gated recurrent units (GRUs) [42–44], are widely employed among deep learning models to process sequence data. These models process batches of data sequentially using a gradient descent algorithm to optimize the unknown model parameters. The gradient information of the model parameters is updated by back-propagation through time [45]. However, due to the sequential processing of input data and back-propagation through time, they suffer from some limitations, especially when dealing with datasets with long dependencies. The training process of LSTM and GRU models also suffers from gradient vanishing and explosion. Though some architectural modifications and training techniques can help LSTM and GRU to alleviate the gradient-related problems to some extent, the effectiveness and efficiency of RNN-based models may still be compromised [46]. Furthermore, it is possible to apply models like Convolutional Neural Network (CNN) to conduct time-series analysis.

On the other hand, the Transformer [47] is a model that combines various mechanisms, such as attention, embedding, and encoder-decoder structures, in natural language processing. Later, studies improved the Transformer and gradually applied it to TSF, imaging, and other areas, making Transformers progressively their genre. Recent advancements in Transformer-based models have shown substantial progress [12, 48, 49]. The self-attentive mechanism of the Transformer allows for adaptive learning of short-term and long-term dependencies through pairwise (query-key) request interactions. This feature grants the Transformer a significant advantage in learning long-term dependencies on sequential data, enabling the creation of more robust and expansive models [50]. The performance of Transformers on LTSF is impressive, and they have gradually become the current mainstream approach.

The two main tasks of time-series data are forecasting and classification. Forecasting aims to predict real values from given time-series data, while the classification task categorizes given time-series data into one or more target categories. Many advances have been made in time-series Transformers for forecasting [12, 49, 51–59] and classification tasks [1, 60–62]. However, genuine time-series data tends to be noisy and non-stationary, and learning spurious dependencies, lacking interpretability, can occur if time-series-related knowledge is not combined. Thus, challenges still need to be addressed despite the notable achievements in accurate long-term forecasting using Transformer-based models.

Following The Preferred Reporting Items for Systematic Reviews and Meta Analysis (PRISMA) standard, we conducted a systematic review and used digital databases, including Google Scholar, Elsevier, and Springer Link. We used the most pertinent keywords, such as “vision transformer” and “long-term time series forecasting”, to choose the most relevant to our study. The number of articles yielded through Google Scholar, Elsevier, and Springer Link since 2020 was 958, 1143, and 703, respectively, for a total of 2804. We first removed duplicate papers, then evaluated the titles and abstracts of these articles, and further reviewed the full text. A total of 59 articles were included in the study. The summary of search results for research articles is shown in Fig. 2.

In this review, we commence with a comprehensive overview of Transformer architecture in Sect. 2. Section 3 presents Transformer-based architectures for LTSF in recent research. In Sect. 4, we analyze Transformer effectiveness for LTSF. Subsequently, Sect. 5

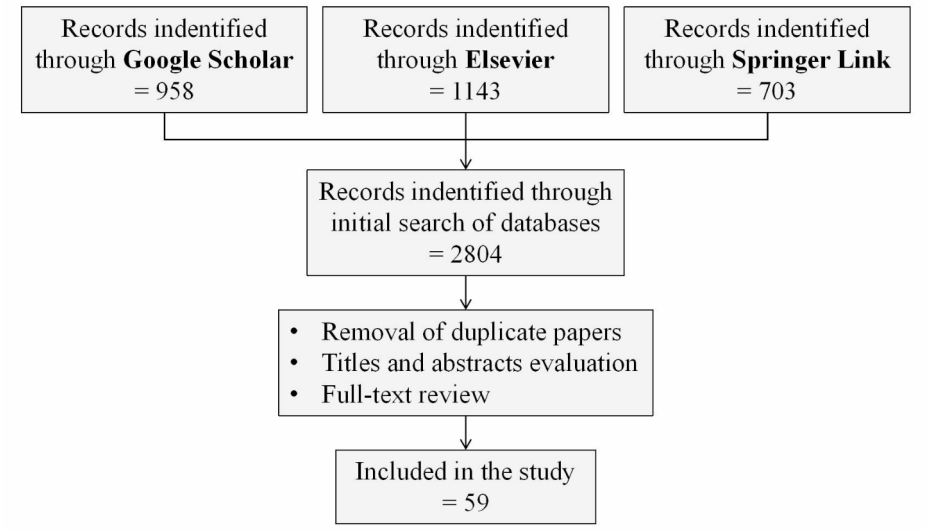

Fig. 2 Article retrieval and selection process based on PRISMA standard

summarizes the public datasets and evaluation metrics in LTSF tasks. Section 6 introduces several training strategies in existing Transformer-based LTSF solutions. Finally, we conclude this review in Sect. 7.

2 Transformer

In this section, we begin by analyzing the inherent mechanics of the Transformer proposed by Vaswani et al. [63] in 2017, to present solutions to the challenge of neural machine translation. Figure 3 shows the Transformer architecture. Subsequently, we delve into the operations within each constituent of the Transformer and the underlying principles that inform these operations. Several variants of the Transformer architecture have been proposed for time-series analysis; however, our discussion in this section is limited to the original architecture [64] [65].

The Transformer network has two parts, the encoder and the decoder, with self-attention for neural sequence transduction. The encoder component encompasses two primary networks: the multi-head attention mechanism and the two-layer feed-forward neural network. It handles symbolic relationships of an input categorization to an incessant relation. The decoder is similar to the encoder, albeit with an extra multi-head attention mechanism that interacts with the encoder output. Unlike the encoder, the decoder comprises three parts of the network structure. The top and bottom segments resemble the encoder, save for a middle section that engages with the encoder's output, referred to as "encoder-decoder attention". The decoder part of the transformer model engenders an output sequence one after the other. Each stage auto-degenerates and exploits the earlier input as supplementary to the next word.

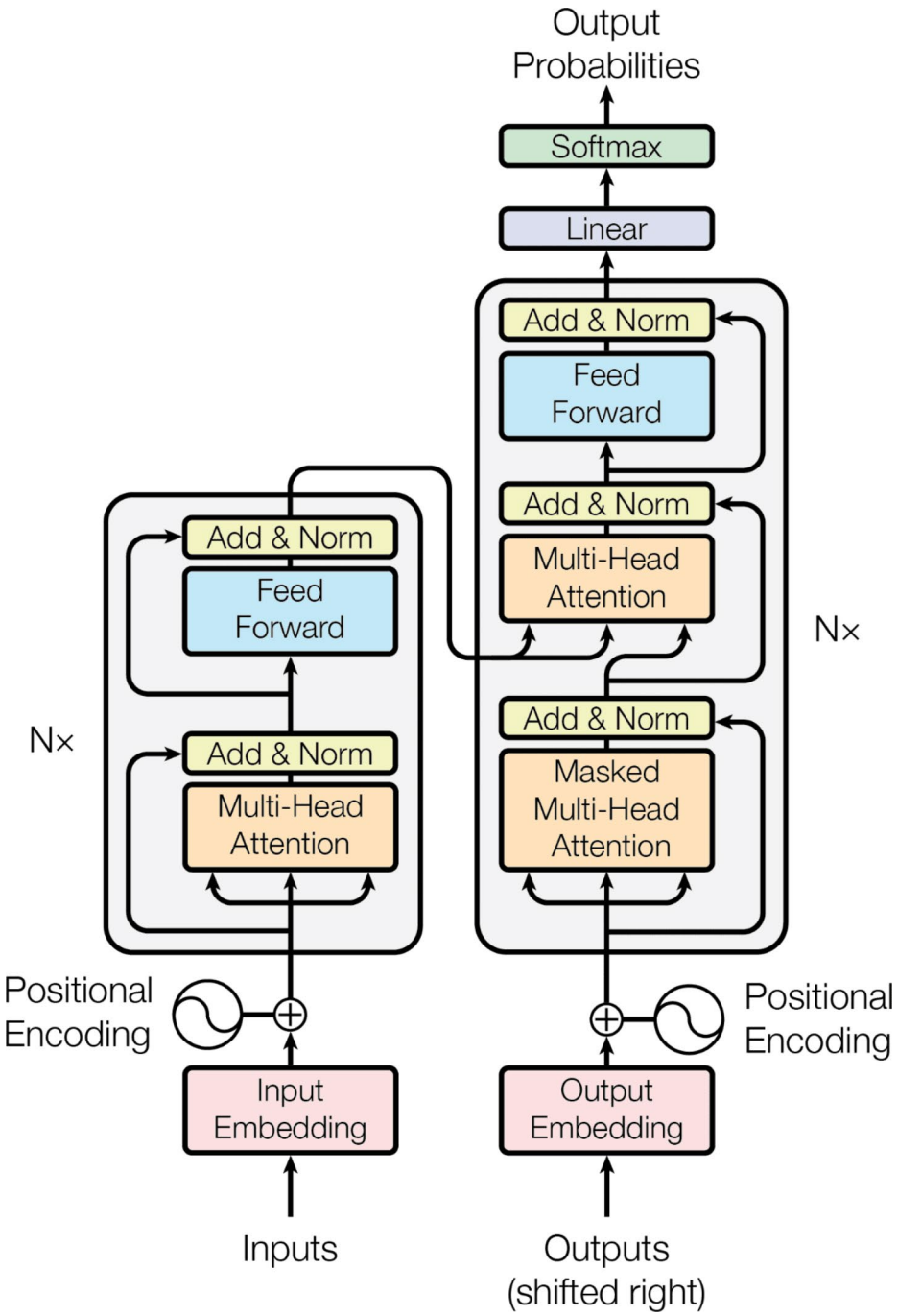

Fig. 3 Schematic diagram of Transformer

2.1 Self-attention

The self-attention mechanism is a process that involves mapping a query and a sequence of key-value pairs to generate a corresponding output vector. The resulting vector is determined by the summation of weights acting on values computed from the query and key. A schematic representation of the self-attention mechanism is depicted in Fig. 4.

As shown in Fig. 4, the core of the self-attention mechanism is to get the attention weights by calculating Q and K and then act on V to get the whole weights and outputs. Q, K, and V are the input sequence's Query, Key, and Value matrices after linear transformation. Concerning the input sequence denoted as X, the parameters Q, K, and V are given by

$$\mathrm{Q} = W_q \mathrm{X}, \mathrm{K} = W_k \mathrm{X}, \text{and } \mathrm{V} = W_v. \tag{1}$$

Q, K, and V are computed by multiplying the input X by three different matrices (but this is only limited to the encoder and decoder encoding process using the self-attention mechanism in their respective inputs; the Q, K, and V in the interaction between the encoder and decoder are referred to otherwise). The computed Q, K, and V can be interpreted as three different linear transformations of the same input to represent its three different states. The

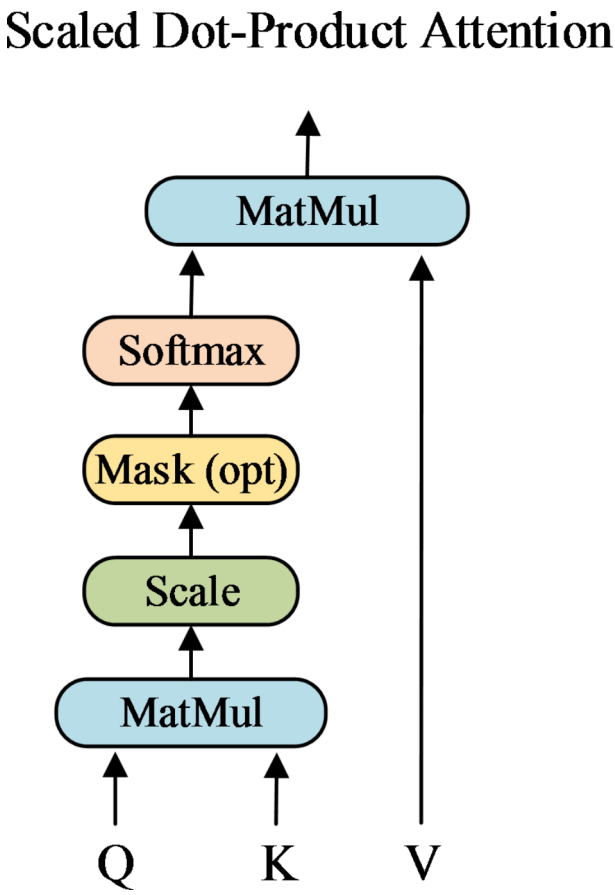

Fig. 4 Schematic diagram of self-attention

weight vectors can be further computed after Q, K, and V are computed. Specifically, for the inputs Q, K, and V, the weight vectors are calculated as:

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\text{V}. \tag{2}$$

The dimension of the query and key is denoted by d_k. The attention for each position is normalized using the softmax function. The formula illustrates that the attention score matrix can be derived by executing a dot product operation between the query and key, followed by division with a scaling factor of $\sqrt{d_k}$. Subsequently, the attention weights for each position are obtained by performing a softmax operation on the attention score matrix. The ultimate self-attention representation is achieved by multiplying the attention weights with the value matrix. The compatibility function employed in this process is a scaled dot product, thus rendering the computation process efficient. Additionally, the linear transformation of the inputs introduces ample expressive power. As illustrated in Fig. 3, the scale process corresponds to the division of d_k in Eq. 2. It is imperative to note that scaling is essential because, for larger d_k, the value obtained after QK^T is excessively large, consequently causing a diminutive gradient after the softmax operation. The diminutive gradient hinders the training of the network and, thus, is not conducive to the overall outcome.

2.2 Multi-head attention

The self-attention mechanism solves the sequential encoding challenge encountered in conventional sequence models. It enables the generation of a final encoded vector that incorporates attention information from multiple positions, achieved through a finite number of matrix transformations on the initial inputs. However, it is worth noting that the model's encoding of positional information may lead to an overemphasis on its position, potentially neglecting the importance of other positions. To address this issue, the Multi-Head Attention mechanism has been introduced.

As shown in Fig. 5, the multi-attention mechanism is a self-attention processing process of multiple groups of the original input sequences, and then each group of self-attention results is spliced together to perform a linear transformation to obtain the final output results. Specifically, its calculation formula is:

$$\begin{aligned}\text{Multi-Head}\left(Q, K, V\right)\text{Concat} &= \left(\text{head}_1, \ldots, \text{head}_h\right)W_O \\ \text{head}_i &= \text{Attention}\left(QW_{Qi}, KW_{Ki}, VW_{Vi}\right).\end{aligned} \tag{3}$$

In this context, the matrices Q, K, and V refer to the input sequences' query, key, and value matrices, respectively, subsequent to linear transformation. The variable h denotes the number of attention heads. Additionally, the weight matrices $W_{Qi}, W_{Ki}, and W_{Vi}$ are utilized to carry out the linear transformation on Q, K, and V. The output weight matrix of the multi-head attention is denoted by the symbol W_O. The computation of a single attention head is denoted by attention in Eq. 2, equivalent to the previously mentioned self-attention mechanism. Each attention head maps the inputs through independent linear transformations and applies the attention mechanism to obtain the representation. The final output of

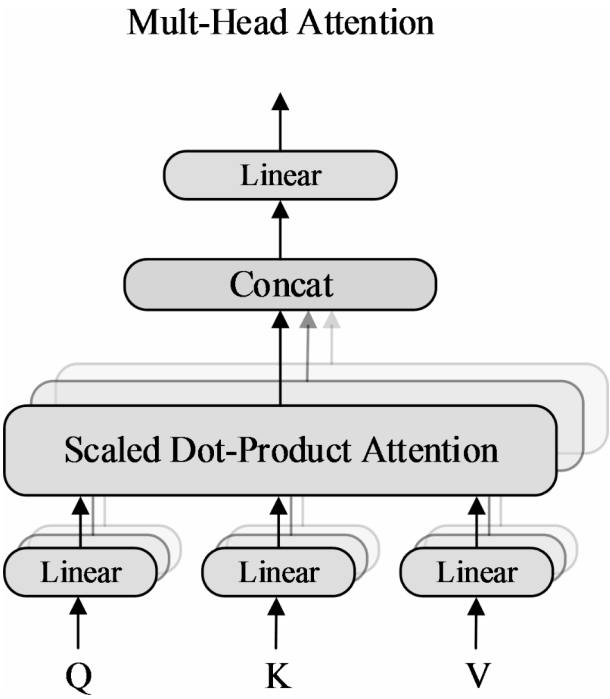

Fig. 5 Multi-head attention

the multi-head attention is obtained by combining the representations of all attention heads and using a linear transformation to the output weight matrix W_O.

3 Transformer-based architectures in LTSF

The design of a network needs to consider the characteristics and nature of problems. In this section, we first analyze the key problems in the LTSF tasks, followed by discussing some popular recent Transformer-based architectures in LTSF tasks.

3.1 LTSF's key problems

LTSF is usually defined as forecasting a more distant future [12, 66]. Given the current status of the existing work, there are two main problems in the field of LTSF: complexity and dependency. LTSF requires processing a large amount of data [67], which may lead to longer training times and require more computational resources [68], as the computational complexity grows exponentially with the length of the sequence. Additionally, storing the entire sequence in memory may be challenging due to the computer's limited memory [69]. This may limit the length of the time series available for prediction [70].

Meanwhile, LTSF models need to have the ability to accurately capture the temporal relationship between past and future observations in a time series [71–73]. Long-sequence time series exhibit long-term dependence [74, 75], challenging the models' ability to capture dependencies [12]. Moreover, LTSF is characterized by inherent periodicity and non-stationarity [76], and thus, LTSF models need to learn a mixture of short-term and long-term

repeated patterns in a given time series [67]. A practical model should capture both repeated ways to make accurate predictions, which imposes more stringent requirements on the prediction model regarding learning dependence.

3.2 Transformer variants

A Transformer [47] mainly captures correlations among sequence data through a self-attention mechanism. Compared with the traditional deep learning architecture, the self-attention mechanism in a Transformer is more interpretable. We have chosen to compare the Transformer-related methods proposed in the last five years. All of these Transformer variants enhance the original Transformer to some extent and can be used for LTSF. Wu et al. [53] introduced the vanilla Transformer to the field of temporal prediction for influenza disease prediction. However, as mentioned above, Transformers have a large computational complexity, leading to high computational costs. Moreover, the utilization of location information is not apparent, the position embedding in the model embedding process is ineffective, and long-distance information cannot be captured. A brief conclusion of recent Transformer-based architectures is given in Table 1.

The time complexity of self-attention computation in a Transformer is initially established at $\mathrm{O}\left(\mathrm{L}^2\right)$, leading to high computational cost. Some subsequent works have been developed to optimize this time complexity and the long-term dependency of Transformer-based models.

The LogSparse Transformer [49] model first introduces Transformer to the field of TSF, making Transformer more feasible for time series with long-term dependencies. LogSparse Transformer allows each time step to be consistent with the previous time step and is selected using an exponential step. It proposed convolutional self-attention by employing causal convolutions to produce queries and keys, reducing time complexity from $\mathrm{O}\left(\mathrm{L}^2\right)$ to $\mathrm{O}\left(L(logL)^2\right)$. The prediction accuracy achieved for fine-grained, long-term dependent time series can be improved in cases with limited memory.

Informer [12] uses the ProbSparse self-attention mechanism, further reducing the computational complexity of the traditional Transformer model to $\mathrm{O}(\mathrm{L}(\log \mathrm{L}))$. At the same time, inspired by dilated convolution in [83] and [84], it also introduced the self-attention distilling operation to remove redundant combinations of value vectors to reduce the total space complexity of the model. In addition, it designed a generative style decoder to produce long sequence outputs with only one forward step to avoid accumulation error. The Informer architecture was tested on various datasets and performed better than models such as Autoregressive Integrated Moving Average (ARIMA) [85], Prophet [86], LSTMa [87], LSTNet [88], and DeepAR [89].

The Autoformer [67] is a simple seasonal trend decomposition architecture with an autocorrelation mechanism working as an attention module. It achieves $\mathrm{O}(\mathrm{L}(\log \mathrm{L}))$ computational time complexity. This deep decomposition architecture embeds the sequence decomposition strategy into the encoder-decoder structure as an internal unit of Autoformer.

In contrast, TCCT [51] designs a CSP attention module that merges CSPNet with a self-attentive mechanism and replaces the typical convolutional layer with an expanded causal convolutional layer, thereby modifying the distillation operation employed by Informer to achieve exponential receptive field growth. In addition, the model develops a penetration

Table 1 Transformer-based architectures

Reference	Technique	Brief information	Time complexity	Lower computational complexity	Higher intersequence dependency
Wu et al. [53]	Transformer	Transformer for LTSF on influenza	$O\left(L^2\right)$		
LogSparse Transformer [49]	LogSparse self-attention+Transformer	Reduce time complexity by convolutional self-attention	$O\left(L(logL)^2\right)$	√	√
AST [56]	GAN+Transformer	Reduce error accumulation by sparse attention with GAN		√	√
SpringNet [77]	Spring attention+Transformer	Spring attention to repeatable long-term dependency fluctuating patterns			√
Lee et al. [78]	Partial correlation-based attention+series-wise multi-resolution	Improve pair-wise comparisons-based attention disadvantages with partial correlation-based attention			√
Informer [12]	ProbSparse self-attention+self-attention distilling+generative style decoder	Sparse and computationally effective	O (L(logL))	√	√
Autoformer [67]	Sequence decomposition+auto-correlation+Transformer	Auto-correlation and sequence decomposition architecture	O (L(logL))	√	√
Pyraformer [70]	Pyramidal attention module+Transformer	Multi-resolution representation with pyramidal attention module	O (L)	√	√
FEDformer [79]	Fourier enhanced+wavelet enhanced+Transformer	Reduce time complexity with frequency domain decomposition based on Autoformer architecture	O (L)	√	√
TCCT [51]	CNN+CSPAttention+Transformer	Reduce computational cost with CSPAttention		√	√
Chu et al. [80]	Autoformer+Informer + Reformer+MLP	Incorporate multiple Transformer variants and meta-learner			√
Quatformer [81]	Learning-to-rotate attention + trend normalization+Transformer	Quaternion architecture with learning-to-rotate attention	O (2cL)	√	√
Muformer [68]	Multi-granularity attention + Transformer+Kullback–Leibler	For multi-sensory domain feature enhancement and multi-headed attentional expression enhancement			√
Triformer [13]	Patch Attention+Transformer	Implement the capture of linear complexity and different temporal dynamic patterns of sequences by a triangular, variable-specific attention architecture	O (L)	√	√
Conformer [82]	Fourier transform+sliding window+Transformer	Extract correlation features of multivariate variables by fast Fourier transform, and improve the operational efficiency of long-period forecasting with a sliding window approach	O (L)	√	√

mechanism for stacking self-attentive blocks to obtain finer information at negligible additional computational costs.

Pyraformer [70] is a novel model based on hierarchical pyramidal attention. By letting the maximum length of the signal traversal path be a constant concerning the sequence length L, it can achieve theoretical O open paren L close paren complexity. Pyraformer conducts both intra-scale and inter-scale attentions, which capture temporal dependencies in an individual resolution and build a multi-resolution representation of the original series, respectively. Similarly, Triformer [13] proposed a triangular, variable-specific attention architecture, which achieves linear complexity through patch attention while proposing a lightweight approach to enable variable-specific model parameters.

FEDformer [79] achieves $O(L)$ linear computational complexity by designing two attention modules that process the attention operation in the frequency domain with the Fourier transform [90] and wavelet transform [91], respectively. Instead of applying Transformer to the time domain, it applies it to the frequency domain, which helps it better expose potential periodic information in the input data.

The Conformer [82] model uses the fast Fourier transform to extract correlation features of multivariate variables. It employs a sliding window approach to improve the operational efficiency of long-period forecasting, sacrificing global information extraction and complex sequence modeling capabilities. Thus, the time complexity is reduced to O open paren L close paren.

To address the problems of long-term dependency, Lin et al. [77] established SpringNet for solar prediction. They proposed a DTW attention layer to capture the local correlations of time-series data, which helps capture repeatable fluctuation patterns and provide accurate predictions. For the same purpose, Chu et al. [80] combined Autoformer, Informer, and Reformer to propose a prediction model based on stacking ensemble learning.

Chen et al. [81] proposed a Quatformer framework in which learning-to-rotate attention introduces learnable period and phase information to describe complex periodic patterns, trend normalization to model normalization of the sequence representation in the hidden layer, and decoupling of the LRA by using the global memory, to efficiently fit multi-periodic complex patterns in the LTSF while achieving linear complexity without loss of prediction accuracy.

To alleviate the problem of redundant information input in LTSF, the Muformer proposed by Zeng et al. [68] enhances the features by inputting the multi-perceptual domain processing mechanism, while the multi-cornered attention head mechanism and the attention head pruning mechanism enhance the expression of multi-head attention. Each of these efforts takes a different perspective on optimizing the parametric part of the model, but a general architecture and component that can reduce the number of required model parameters has not yet emerged.

In addition to the previously mentioned Transformer-based architectures, other architectural modifications have emerged in recent years. For example, the Bidirectional Encoder Representations from Transformers (BERT) [92] model is built by stacking Transformer encoder modules and introducing a new training scheme. Pre-training the encoder modules is task-independent, and decoder modules can be added later and fine-tuned to the task. This scheme allows BERT models to be trained on large amounts of unlabeled data. BERT architecture has inspired many new Transformer models for time-series data [1, 57, 60]. However, compared to NLP tasks, time-series data include various data types [1, 12, 93].

Thus, the pre-training process will have to be different for each task. This task-dependent pre-training contrasts with the NLP tasks, which can start with the same pre-trained models, assuming all tasks are based on the same language semantics and structure.

Generative adversarial networks (GANs) consist of the generator and the discriminator, learning adversarially from each other. The generator-discriminator learning principle has been applied to the time-series forecasting task [56]. The authors use a generative adversarial encoder-decoder framework to train a sparse Transformer model for time-series forecasting, solving the problem of being unable to predict long series due to error accumulation. The adversarial training process improves the model's robustness and generalization ability by directly shaping the output distribution of the network to avoid error accumulation through one-step-ahead inference.

TranAD [94] applied GAN-style adversarial training with two Transformer encoders and two Transformer decoders to gain stability. As a simple Transformer-based network tends to miss slight deviations of anomaly, an adversarial training procedure can amplify reconstruction errors.

TFT [54] designs a multi-horizon model with static covariate encoders, a gating feature selection module, and a temporal self-attention decoder. It encodes and selects valuable information from various covariates information to perform forecasting. It also preserves interpretability by incorporating global and temporal dependencies and events. SSDNet [95] combines the Transformer with state space models (SSM), which use the Transformer part to learn the temporal pattern and estimate the SSM parameters; the SSM parts perform the seasonal-trend decomposition to maintain the interpretable ability. While MT-RVAE [96] combines the Transformer with Variational AutoEncoder (VAE), it focuses on data with few dimensions or sparse relationships. A multi-scale Transformer is designed to extract different levels of global time-series information. AnomalyTrans [60] combines Transformer and Gaussian prior association to make rare anomalies more distinguishable. Prior association and series association are modeled simultaneously. The minimax strategy optimizes the anomaly model to constrain the prior and series associations for more distinguishable association discrepancies.

GTA [3] contains a graph convolution structure to model the influence propagation process. Replacing vanilla multi-head attention with a multi-branch attention mechanism combines global-learned attention, multi-head attention, and neighborhood convolution. GTN [62] applies a two-tower Transformer, with each tower working on time-step-wise attention and channel-wise attention, respectively. A learnable weighted concatenation is used to merge the features of the two towers. Aliformer [57] makes the time-series sales forecasting using knowledge-guided attention with a branch to revise and denoise the attention map.

In addition, some researchers have made corresponding network improvements for specific applications. First, in the transportation application, spatiotemporal graph Transformer [97] proposes an attention-based graph convolution mechanism for learning a more complex temporal-spatial attention pattern applying to pedestrian trajectory prediction. Traffic Transformer [55] designs an encoder-decoder structure using a self-attention module to capture the temporal-temporal dependencies and a graph neural network (GNN) module to capture the spatial dependencies. Spatial-temporal Transformer networks introduced a temporal Transformer block to capture the temporal dependencies and a spatial Transformer block to assist a graph convolution network to capture more spatial-spatial dependencies [98].

There are also applications for event prediction. Event forecasting or prediction aims to predict the times and marks of future events given the history of past events, which is often modeled by temporal point processes (TPP) [6]. Self-attentive Hawkes process (SAHP) [7] and Transformer Hawkes process (THP) [8] adopt Transformer encoder architecture to summarize the influence of historical events and compute the intensity function for event prediction. They modify the positional encoding by translating time intervals into sinusoidal functions to utilize interval between events. Later, a more flexible model named attentive neural datalog through time (ANDTT) [99] was proposed to extend SAHP/THP schemes by embedding all possible events and times with attention.

4 Transformer effectiveness for LTSF

Is Transformer effective in the time series forecasting domain? The response we provide is affirmative. Since the publication of Zeng's scholarly article, "Are Transformers effective for time series forecasting?"[100], the feasibility of utilizing Transformer models for time series forecasting has emerged as a significant subject of scholarly discourse. This is particularly noteworthy as a straightforward model emerged victorious over a considerably intricate Transformer model, thus prompting a substantial academic discourse. Zeng claimed that the Transformer-based models are not effective in time series forecasting. They compare the Transformer-based models with a simple linear model, DLinear, which uses the decomposition layer structure in Autoformer and which DLinear claims outperforms the Transformer-based models. A Transformer with different positional and temporal embeddings retains very limited temporal relationships. It is prone to overfitting on noisy data, whereas a linear model can be modeled in a natural order and with fewer parameters can avoid overfitting. However, Nie [101] presents a novel solution to tackle the loss of temporal information induced by the self-attention mechanism. This approach is rooted in the Transformer time-series prediction and involves transforming the time-series data into a patch format akin to that of Vision Transformer. This conversion preserves the localization of the time series, with each patch serving as the smallest unit for Attention computation. The findings in Table 2 demonstrate that research focused on Transformer-based time-series prediction underscores the significance of integrating temporal information to improve the model's prediction performance.

A straightforward linear model may have its advantages in specific circumstances; however, it may need to be more capable of effectively handling extensive time series information on the same level as a more intricate model, such as the Transformer. In summary, the Transformer model still needs to be updated in time series forecasting. Nonetheless, having abundant training data to unlock its immense potential is crucial. Unfortunately, there is currently a scarcity of publicly available datasets that are sufficiently large for time series forecasting. Most existing pre-trained time-series models use public datasets like Traffic and Electricity. Despite these benchmark datasets serving as the foundation for developing time series forecasting, their limited size and lack of generalizability pose significant challenges for large-scale pre-training. Thus, in the context of time-series prediction, the most pressing matter is the development of expansive and highly generalized datasets (similar to ImageNet in computer vision). This crucial step will undoubtedly propel the advancement of time-series analysis and training models while enhancing the capacity of training mod-

Table 2 Multivariate long-term forecasting results on electricity dataset

Models		PatchTST/64		PatchTST/42		DLinear		FEDformer		Autoformer		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
prediction length	96	0.129	0.222	0.130	0.222	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393
	192	0.147	0.240	0.148	0.240	0.153	0.249	0.197	0.311	0.211	0.324	0.327	0.417
	336	0.163	0.259	0.167	0.261	0.169	0.267	0.213	0.328	0.214	0.327	0.333	0.422
	720	0.197	0.290	0.202	0.291	0.203	0.301	0.233	0.344	0.236	0.342	0.351	0.427

els in time-series prediction. Additionally, this development underscores the Transformer model's effectiveness in successfully capturing long-term dependencies within a sequence while maintaining superior computational efficiency and a more comprehensive feature representation capability.

On the other hand, the Transformer's effectiveness is reflected in Large Language Models (LLMs). LLMs are powerful Transformer-based models, and numerous previous studies have shown that Transformer-based models are capable of learning potentially complex relationships among textual sequences [102, 103]. It is reasonable to expect LLMs to have the potential to understand complex dependencies among numeric time series augmented by temporal textual sequences.

The current endeavor for time series LLMs encompasses two primary strategies. One approach involves creating and preliminary training a fundamental, comprehensive model specifically tailored for time series. This model can be subsequently fine-tuned to cater to various downstream tasks. This path represents the most rudimentary solution, drawing upon a substantial volume of data and imbuing the model with time-series-related knowledge through pre-training. The second strategy involves fine-tuning based on the LLM framework, wherein corresponding mechanisms are devised to adapt the time series, enabling application to existing language models. Consequently, this facilitates processing diverse time-series tasks using the pre-existing language models. This path poses challenges and necessitates transcending the original language model.

5 Public datasets and evaluation metrics

In this section, we summarize some typical applications and relevant public LTSF datasets. We also discuss the prediction evaluation metrics in LTSF.

5.1 Common applications and public datasets

5.1.1 Finance

LTSF is commonly used in finance to predict economic cycles [104], fiscal cycles, and long-term stock trends [105]. LTSF can predict future trends and stock price fluctuations in the stock market, helping investors develop more accurate investment strategies. In financial planning, LTSF can predict future economic conditions, such as income, expenses, and profitability, to help individuals or businesses better plan their financial goals and capital operations [106]. In addition, LTSF can predict a borrower's repayment ability and credit risk [107] or predict future interest rate trends to help financial institutions conduct loan risk assessments for better monetary and interest rate policies. We summarized the open-source LTSF datasets in the finance field in recent years in Table 3.

5.1.2 Energy

In the energy field, LTSF is often used to assist in developing long-term resource planning strategies [118]. It can help companies and governments forecast future energy demand to better plan energy production and supply. It can also help power companies predict future

Table 3 Finance LTSF dataset

Dataset	Reference	Data information	Min-granularity
Gold prices	[108]	Daily gold prices from 2014.1 to 2018.4, including minimum, mean, maximum, median, standard deviation, skewness, and kurtosis http://finance.yahoo.com	1 day
GEFCom2014 Electricity Price	[109]	Dataset consists of electricity load forecasting, electricity price forecasting, wind, and solar power generation https://www.dropbox.com/s/pqenrr2mcvl0hk9/GEFCom2014.zip? dl=0	1 h
Exchange-rate	[67, 110–112]	Daily exchange rates from Australia, the United Kingdom, Canada, Switzerland, China, Japan, New Zealand, and Singapore between 1990 and 2016 https://github.com/laiguokun/multivariate-time-series-data	1 day
S&P 500	[113]	Daily S&P 500 index from 2001.1 to 2017.5 http://finance.yahoo.com	1 day
Shanghai composite	[113]	Daily SSE indices from 2005.1 to 2017.6 http://finance.yahoo.com	1 day
S&P 500 stocks	[114]	505 common stocks traded on the American stock exchange, recording historical daily stock prices for all companies currently included in the S&P 500 index from 2013.2 to 2018.2 https://www.kaggle.com/camnugent/sandp500	1 day
CRSP's stocks	[115]	Data on individual stock returns and prices, S&P 500 index returns, industry categories, number of shares outstanding, ticker symbols, exchange codes, and trading volume http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html	1 day
Gas station revenue	[73]	Daily revenue of five gas stations from 2015.12 to 2018.12 https://github.com/bighuang624/DSANet/tree/master/data	1 day
Finance Japan	[116]	Dataset collected by the Ministry of Finance of Japan that records general partnerships, limited partnerships, limited liability companies, and joint stock companies from 2003.1 to 2016.12 https://www.mof.go.jp/english/pri/reference/ssc/outline.htm	4 months
Stock opening prices	[117]	Daily opening prices for 50 stocks in 10 sectors in Financial Yahoo between 2007 and 2016 https://github.com/z331565360/State-Frequency-Memory-stock-prediction	1 day

power generation, ensuring a sufficient and stable power supply [119]. In addition, LTSF can help governments and enterprises to develop energy policy planning or manage the energy supply chain [120]. These applications can help enterprises and governments better plan, manage, reduce risks, improve efficiency, and realize sustainable development. We summarized the energy field's open-source datasets in recent years in Table 4.

5.1.3 Transportation

In urban transportation, LTSF can help urban traffic management predict future traffic flow [123] for better traffic planning and management. It can also be used to predict future traffic congestion [124], future traffic accident risks, and traffic safety issues [125] for better traffic safety management and accident prevention. We summarized the open-source datasets in the transportation field in recent years in Table 5.

5.1.4 Meteorology and medicine

The application of LTSF in meteorology mainly focuses on predicting long-term climate trends. For example, LTSF can be used to predict long-term climate change [133], providing a scientific basis for national decision-making in response to climate change. It can also issue early warnings for natural climate disasters [134] to mitigate potential hazards to human lives and properties. In addition, LTSF can predict information such as sea surface temperature and marine meteorology for the future [135], providing decision support for industries such as fisheries and marine transportation. We summarized the open-source datasets in the meteorology and medicine fields in recent years in Tables 6 and 7, respectively.

In the medical field, LTSF can be applied to various stages of drug development. For example, predicting a drug's toxicity, pharmacokinetics, pharmacodynamics, and other parameters helps researchers optimize the drug design and screening process [137]. In addition, LTSF can predict medical needs over a certain period [138]. These predictions can be used to allocate and plan medical resources rationally.

5.2 Evaluation metrics

In this section, we discuss prediction performance evaluation metrics in the field of TSF. According to [141], the prediction accuracy metrics can be divided into three groups: scale-dependent, scale-independent, and scaled error metrics, based on whether the evaluation metrics are affected by the data scale and how the data scale effects are eliminated.

Let Y_t denote the observation at time t (t=1,…, n) and F_t denote the forecast of Y_t. Then the forecast error is defined as $e_t = Y_t - F_t$.

5.2.1 Scale-dependent measures

Scale-dependent measures are the most widely used evaluation metrics in forecasting, whose data scales depend on the data size of the original data. This type of metric is computationally simple. These are useful when comparing different methods applied to the same datasets but should not be used, for example, when comparing across datasets with different scales.

Table 4 Energy LTSF dataset

Dataset	Reference	Data information	Min-granularity
Power consumption	[117]	The electricity consumption of a household, including voltage, electricity consumption, and other characteristics from 2006.12 to 2010.11 https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption	1 min
Solar energy	[49, 56, 110, 111, 121]	The highest solar power production from 137 photovoltaic plants in Alabama in 2006 https://www.nrel.gov/grid/solar-power-data.html	5 min
electricity	[56, 67, 110, 111, 121]	The electricity consumption of 321 customers between 2011 and 2014 https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014	15 min
wind	[49, 56]	Hourly estimates of energy potential as a percentage of the maximum output of power plants for a European region between 1986 and 2015 https://www.kaggle.com/sohier/30-years-ofeuropean-wind-generation	1 h
ETT	[12, 67]	The load and oil temperature of power transformers from 2016.7 to 2018.7 https://github.com/zhouhaoyi/ETDataset	15 min
sanyo	[77]	Daily solar power generation data from two photovoltaic plants in Alice Springs, Northern Territory, and Australia from 2011.1 to 2017.1 http://dkasolarcentre.com.au/source/alicesprings/dka-m4-b-phase	1 day
hanergy	[77]	Daily solar power generation data from two photovoltaic plants in Alice Springs, Northern Territory, and Australia from 2011.1 to 2016.12 https://dkasolarcentre.com.au/source/alicesprings/dka-m16-b-phase	1 day
power grid data	[122]	Grid data of State Grid Shanghai Municipal Electric Power Company from 2014.1 to 2015.2	1 day

The most commonly used scale-dependent measures are based on the absolute error or squared errors:

$$\text{Mean Square Error (MSE)} = \text{mean}\left(e_t^2\right) \tag{7}$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\text{MSE}} \tag{8}$$

$$\text{Mean Absolute Error (MAE)} = \text{mean}\left(|e_t|\right) \tag{9}$$

$$\text{Median Absolute Error (MdAE)} = \text{median}\left(|e_t|\right) \tag{10}$$

Historically, the RMSE and MSE have been popular because of their theoretical relevance in statistical modeling. The RMSE is effective for its simplicity and close relationship with statistical modeling. It can give the same value as the forecast error variance for unbiased forecasting. However, they are more sensitive to outliers than MAE or MdAE [142]. The MAE better reflects the actual error situation than the RMSE.

Table 5 Transportation LTSF dataset

Dataset	Reference	Data information	Min-granularity
Paris metro line	[25]	The passenger flow on Paris metro lines 3 and 13 between 2009 and 2010 http://www.neural-forecasting-competition.com/	1 h
PeMS03, PeMS04, PeMS07, PeMS08,	[56, 67, 110, 111, 121, 126–128]	traffic flow data http://pems.dot.ca.gov/	30s
Birmingham Parking	[114]	The parking lot ID, parking lot capacity, parking lot occupancy, and update time for 30 parking lots operated by Birmingham National Car Park from 2016.10 to 2016.12 http://archive.ics.uci.edu/ml/datasets/parking+birmingham	30 min
METR-LA	[110, 126]	Traffic information collected by loop detectors on Los Angeles County freeways from 2012.3 to 2012.6 https://drive.google.com/drive/folders/10FOTa6HXPqX8Pf5WRoRwcFnW9BrNZEIX	5 min
PEMS-BAY	[110, 126]	Traffic speed readings from 325 sensors collected by PeMS, the California Transit Agency Performance Measurement System from 2017.1 to 2017.5 https://drive.google.com/drive/folders/10FOTa6HXPqX8Pf5WRoRwcFnW9BrNZEIX	30s
SPMD	[129]	The driving records of approximately 3,000 drivers in Ann Arbor, Michigan, from 2015.5 to 2015.10 https://github.com/ElmiSay/DeepFEC	1 h
VED	[129]	The fuel and energy consumption of various personal vehicles operating under different realistic driving conditions in Michigan, US, from 2017.11 to 2018.11 https://github.com/ElmiSay/DeepFEC	1 h
England	[127]	National average speeds and traffic volumes derived from UK freeway traffic data from 2014.1 to 2014.6 http://tris.highwaysengland.co.uk/detail/trafficflowdata	15 min
TaxiBJ+	[130]	The distribution and trajectory of more than 3,000 cabs in Beijing https://www.microsoft.com/enus/research/publication/deep-spatio-temporal-residualnetworks-for-citywide-crowd-flows-prediction	30 min
BikeNYC	[130]	Trajectory data taken from the NYC Bike system in 2014, from April 1 to Sept. 30. Trip data includes trip duration, starting and ending station IDs, and start and end times https://www.microsoft.com/enus/research/publication/deep-spatio-temporal-residualnetworks-for-citywide-crowd-flows-prediction	1 h
HappyValley	[131]	The hourly population density of popular theme parks in Beijing from 2018.1 to 2018.10 heat.qq.com	1 h
NYC Taxi	[132]	Details of every cab trip in New York City from 2009.1 to 2016.6	1 h

5.2.2 Scale-independent measures

Scale-independent measures are evaluation metrics not affected by the size of the original data. They can be divided more specifically into three subcategories: measures based on percentage errors, measures based on relative errors, and relative measures.

The percentage error is $p_t = 100e_t/Y_t$. The most commonly used measures are:

$$\text{Mean Absolute Percentage Error (MAPE)} = \text{mean}(|\text{p}_\text{t}|) \tag{11}$$

$$\text{Median Absolute Percentage Error (MdAPE)} = \text{median}(|\text{p}_\text{t}|) \tag{12}$$

$$\text{Root Mean Square Percentage Error (RMSPE)} = \sqrt{\text{mean}(\text{p}_\text{t}^2)} \tag{13}$$

$$\text{Root Median Square Percentage Error (RMdSPE)} = \sqrt{\text{median}(\text{p}_\text{t}^2)} \tag{14}$$

Percentage errors have the advantage of being scale-independent and so are frequently used to compare forecast performance across different datasets. However, these measures have the disadvantage of being infinite or undefined if $Y_t = 0$ for any t in the period of interest and have an extremely skewed distribution when any value of Y_t is close to zero. The MAPE and MdAPE also have the disadvantage of putting a heavier penalty on positive errors than negative errors. Measures based on percentage errors are often highly skewed, and, therefore, transformations (such as logarithms) can make them more stable [143].

An alternative scaling method is dividing each error by the error obtained using another standard forecasting method. Let $r_t = e_t/e_t^*$ denote the relative error, where e_t^* is the forecast error obtained from the benchmark method. Usually, the benchmark method is the random walk where F_t is equal to the last observation.

$$\text{Mean Relative Absolute Error (MRAE)} = \text{mean}(|\text{r}_\text{t}|) \tag{15}$$

$$\text{Median Relative Absolute Error (MdRAE)} = \text{median}(|\text{r}_\text{t}|) \tag{16}$$

$$\text{Geometric Mean Relative Absolute Error (GMRAE)} = \text{gmean}(|\text{r}_\text{t}|) \tag{17}$$

A serious deficiency of relative error measures is that e_t^* can be small. In fact, r_t has infinite variance because e_t^* has a positive probability density at 0. Using "winsorizing" can trim extreme values, which will avoid the difficulties associated with small values of e_t^* [144], but adds some complexity to the calculation and a level of arbitrariness as the amount of trimming must be specified.

Rather than use relative errors, one can use relative measures. For example, let MAE_b denote the MAE from the benchmark method. Then, a relative MAE is given by

$$\text{relMAE} = \text{MAE}/\text{MAE}_\text{b}. \tag{18}$$

An advantage of these methods is their interpretability. However, they require several forecasts on the same series to compute MAE (or MSE).

Table 6 Meteorology LTSF dataset

Dataset	Reference	Data information	Min-granularity
Beijing PM2.5	[136]	Hourly PM2.5 data and associated meteorological data for Beijing from 2010.1 to 2014.12 https://archive.ics.uci.edu/ml/datasets.html	1 h
Hangzhou temperature	[113]	Daily average temperature of Hangzhou from 2011.1 to 2017.1 http://data.cma.cn/data/	1 day
WTH	[67]	Weather conditions throughout 2020 https://www.bgc-jena.mpg.de/wetter/	10 min
USHCN	[34]	Continuous daily meteorological records from 1887 to 2009 https://www.ncdc.noaa.gov/ushcn/introduction.	1 day
KDD-CUP	[34]	PM2.5 measurements from 35 monitoring stations in Beijing from 2017.1 to 2017.12 https://www.kdd.org/kdd2018/kdd-cup.	1 h
US	[128]	Weather datasets from 2012 to 2017 from 36 weather stations in the US https://www.kaggle.com/selfishgene/historical-hourly-weather-data.	1 h

Table 7 Medicine LTSF dataset

Dataset	Reference	Data information	Min-granularity
ILI	[53, 67]	Data on patients with influenza-like illness recorded weekly by the Centers for Disease Control and Prevention from 2002 to 2021 https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html.	1 week
COVID-19	[139]	Daily data on confirmed and recovered cases collected from 2020.1 to 2020.6 in Italy, Spain, France, China, US, and Australia https://github.com/CSSEGISandData/COVID-19	1 day
2020 OhioT1DM	[140]	Eight weeks of continuous glucose monitoring, insulin, physiological sensor, and self-reported life event data for each of 12 patients with type 1 diabetes in 2020 http://smarthealth.cs.ohio.edu/OhioT1DM-dataset.html	5 min
MIMIC-III	[34]	A public clinical dataset with over 58,000 admission records from 2001 to 2012 http://mimic.physionet.org	1 h

5.2.3 Scaled errors

Scaled errors were first proposed in [141] and can be used to eliminate the effect of data size by comparing the prediction results obtained with the underlying method (usually the native method). The following scaled error is commonly used:

$$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|}. \tag{19}$$

Therefore, The Mean Absolute Scaled Error is simply $\mathrm{MASE} = \mathrm{mean}(|\mathrm{q_t}|)$.

The denominator can be considered as the average error of the native predictions made one step ahead in the future. If MASE>1, the experimental method under evaluation is

worse than the native prediction, and vice versa. Similar to MASE, MASE is calculated using the mean, making it more susceptible to outliers, while MdASE computed using the median has stronger robustness and validity. However, such metrics can only reflect the results of comparison with the primary method and cannot visualize the error of the prediction results.

6 Training strategies

Recent Transformer variants introduce various time-series features into the models for improvements [67, 70]. In this section, we summarize several training strategies of existing Transformer-based models for LTSF.

6.1 Preprocessing and embedding

In the preprocessing stage, normalization with zero mean is often applied in time-series tasks. Moreover, seasonal-trend decomposition is a standard method to make raw data more predictable [145, 146], first proposed by Autoformer [67]. It also uses a moving average kernel on the input sequence to extract the trend-cyclical component of the time series. The seasonal component differs between the original sequence and the trend component. FEDformer [79] further proposed a mixture of experts' strategies to mix the trend components extracted by moving average kernels with various kernel sizes.

The self-attentive layer in the Transformer architecture cannot preserve the positional information of the time series. However, local location information or the ordering of the time series is essential. Furthermore, global time information is also informative, such as hierarchical timestamps (weeks, months, years) and agnostic timestamps (holidays and events) [12]. To enhance the temporal context of the time-series input, a practical design is to inject multiple embeddings into the input sequence, such as fixed positional coding and learnable temporal embeddings. Additionally, the introduction of temporal embeddings accompanied by temporal convolutional layers [49] or learnable timestamps [67] has been proposed as an effective means further to enhance the temporal context of the input data.

6.2 Iterated multi-step and direct multi-step

The time series forecasting task is to predict the values at the T future time steps. When $T>1$, iterated multi-step (IMS) forecasting [147] learns a single-step forecaster and iteratively applies it to obtain multi-step predictions. Alternatively, direct multi-step (DMS) forecasting [148] optimizes the multi-step forecasting objective simultaneously. The variance of the IMS predictions is smaller due to the autoregressive estimation procedure compared to DMS forecasting but is inevitably subject to the error accumulation effects. Therefore, IMS forecasting is more desirable when highly accurate single-step forecasters exist, and T is relatively small. In contrast, DMS forecasting produces more accurate forecasts when unbiased single step forecast models are challenging to obtain or when T is large.

Applying the vanilla Transformer model to the LTSF problem has some limitations, including the quadratic time/memory complexity with the original self-attention scheme and error accumulation caused by the autoregressive decoder design. Alternative Trans-

former variants have been developed to overcome these challenges, each employing distinct strategies. For instance, LogTrans [49] introduces a dedicated decoder for IMS forecasting, while Informer [12] leverages a generative-style decoder. Additionally, Pyraformer [70] incorporates a fully connected layer that concatenates spatiotemporal axes as its decoder. Autoformer [67] adds the two refined decomposition features of the trend-cyclical components and the stacked autocorrelation mechanism of the seasonal component to obtain the final prediction results. Similarly, FEDformer [79] applies a decomposition scheme and employs the proposed frequency attention block in deciphering the final results.

7 Conclusion

Transformer architecture has been found to be applicable to solving various time-series tasks. The Transformer architecture based on self-attention and positional encoding offers better or similar performance as RNNs and variants of LSTMs/GRUs. However, it is more efficient in computing time and overcomes other shortcomings of RNNs/LSTMs/GRUs.

In this paper, we summarized the application of the Transformer on LTSF. First, we have provided a thorough examination of the fundamental structure of the Transformer. Subsequently, we analyzed and summarized the advantages of Transformer on LTSF tasks. Given that the Transformer encounters intricacies and interdependencies when confronting LTSF tasks, numerous adaptations have been introduced to the original architectural framework, thus equipping Transformers with the capacity to handle LTSF tasks effectively. This architectural augmentation, however, brings certain challenges during the training process. To address this, we have incorporated a compendium of best practices that facilitate the practical training of Transformers. Additionally, we have collected abundant resources on TSF and LTSF, including datasets, application fields, and evaluation metrics.

In summary, our comprehensive review examines recent advancements in Transformer-based architecture in LTSF and imparts valuable insights to researchers seeking to improve their models. The Transformer architecture is renowned for its remarkable modeling capacity and aptitude for capturing long-term dependencies. However, it encounters challenges regarding time complexity when applied to LTSF tasks. While efforts to reduce complexity may inadvertently lead to the loss of certain interdependencies between data points, thereby compromising prediction accuracy. Consequently, the amalgamation of various techniques within a compound model, leveraging the strengths of each, emerges as a promising avenue for future research in Transformer-based LTSF models. This paves the way for innovative model designs, data processing techniques, and benchmarking approaches to tackle the intricate LTSF problems. In future research, progressive trending and seasonal decomposition mechanisms can be introduced as multiple cycles and trends are hidden and repeated among the data. At the same time, the Transformer-based models have some inherent limitations in LSTF, such as time complexity. Transformer-based models may unavoidably lose some of the dependencies between data points when reducing the complexity of self-attention computations in situations with many data feature variables with complex correlations, resulting in reduced prediction accuracy. Therefore, Transformer-based models are not suitable for all LSTF tasks. Also, pre-trained Transformer models for different tasks in time series and more architecture-level designs for Transformers can be investigated in depth in the future. Notably, researchers have recently explored the integration of Large Language Models (LLMs)

in time series forecasting, wherein LLMs exhibit the capability to generate forecasts while offering human-readable explanations for predictions, outperforming traditional statistical models and machine learning approaches. These encouraging findings present a compelling impetus for further exploration, aiming to enhance the precision, comprehensibility, and transparency of forecasting results.

Acknowledgements This work was supported by the Project of the Educational Commission of Guangdong Province of China (2022ZDJS113) and the Natural Science Foundation of Top Talent of SZTU (GDRC20221).

Author contributions L.S. and X.Z. wrote the manuscript, conceived of the presented idea and designed the manuscript. R.L., X.W carried out the collections. H.Z. and B.H. supervised the project. All authors approved the manuscript.

Data availability Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Yuan Y, Lin L (2021) Self-Supervised Pretraining of Transformers for Satellite Image Time Series Classification. IEEE J Sel Top Appl Earth Observations Remote Sens 14:474–487.
2. Zerveas G et al (2021) A Transformer-based Framework for Multivariate Time Series Representation Learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, pp 2114–2124
3. Chen Z et al (2021) Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. IEEE Internet of Things Journal 9(12):9179–9189
4. Meng H et al (2019) Spacecraft Anomaly Detection via Transformer Reconstruction Error. In: Proceedings of the International Conference on Aerospace System Science and Engineering 2019, Lecture Notes in Electrical Engineering, 622:351–362
5. Ruff L et al (2021) A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE 109(5):756–795
6. Shchur O et al (2021) *Neural Temporal Point Processes: A Review.* arXiv preprint http://arxiv.org/abs/2104.03528
7. Zhang Q et al (2020) Self-attentive Hawkes process. In: International conference on machine learning, PMLR, pp 11183–11193
8. Zuo S et al (2020) *Transformer Hawkes Process*. In: International conference on machine learning, PMLR, pp 11692–11702
9. Esling P, Agon C (2012) Time-series data mining. ACM-CSUR 45(1):1–34
10. Lim B, Zohren S (2021). Time-series forecasting with deep learning: a survey. Philos T Roy Soc A 379(2194):20200209

11. Torres JF et al (2021) Deep Learning for Time Series Forecasting: A Survey. Big Data 9(1):3–21
12. Zhou H et al (2020) *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting.* arXiv preprint http://arxiv.org/abs/2012.07436
13. Cirstea RG et al (2022) *Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting-Full Version.* arXiv preprint http://arxiv.org/2204.13767
14. Liang Y et al (2018) *GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction.* In: International Joint Conference on Artificial Intelligence, pp 3428–3434
15. Gao J et al (2009) Denoising Nonlinear Time Series by Adaptive Filtering and Wavelet Shrinkage: a comparison. IEEE Signal Process Lett 17(3):237–240
16. Rojo-Alvarez JL et al (2004) Support vector method for robust ARMA system identification. IEEE Trans Signal Process 52(1):155–164
17. Hong T, Fan S (2016) Probabilistic electric load forecasting: a tutorial review. Int J Forecast 32(3):914–938
18. Miller C et al (2020) The ASHRAE Great Energy Predictor III competition: Overview and results. Sci Technol Built Environ 26:1427–1447
19. Shao H, Soong BH (2016) Traffic flow prediction with long short-term memory networks (LSTMs). In: 2016 IEEE region 10 conference (TENCON), IEEE, pp 2986–2989
20. Yule GU (1927) On a method of investigating periodicities in distributed Series, with special reference to Wolfer's sunspot numbers. Phil Trans R Soc Lond A 226:267–298
21. Walker GT (1931) On periodicity in series of related terms. Proc Royal Soc Lond Ser Containing Papers Math Phys Character 131(818):518–532
22. Rojas I et al (2008) Soft-computing techniques and ARMA model for time series prediction. Neurocomputing 71(4–6):519–537
23. Box GEP, Pierce DA (1970) Distribution of residual in Autoregressive-Integrated moving average Time Series. J Am Stat Assoc 65(332):1509–1526
24. Marple SL Jr, Carey WM (1998) *Digital Spectral Analysis with Applications.* J Acoust Soc Am, 86(5):2043
25. Wang Q et al (2020) A deep granular network with adaptive unequal-length granulation strategy for long-term time series forecasting and its industrial applications. Artif Intell Rev 53(7):5353–5381
26. Farnoosh A et al (2020) *Deep Switching Auto-Regressive Factorization:Application to Time Series Forecasting.* arXiv preprint http://arxiv.org/2009.05135
27. McDonald DJ et al (2012) Nonparametric Risk Bounds for Time-Series Forecasting. J Mach Learn Res 18(32):1–40
28. Wen Q et al (2018) *RobustSTL: A robust seasonal-trend decomposition algorithm for long time series.* In: Proceedings of the AAAI conference on artificial intelligence 33(1):5409–5416
29. Yang X et al (2017) Long-term forecasting of time series based on linear fuzzy information granules and fuzzy inference system. Int J Approximate Reasoning 81:1–27
30. Cortes C, Vapnik V (1995) Support-Vector Networks. Mach Learn 20(3):273–297
31. Freund Y (1995) Boosting a weak learning algorithm by Majority. Inf Comput 121(2):256–285
32. Liu Z et al (2018) *A Flexible Forecasting Framework for Hierarchical Time Series with Seasonal Patterns: A Case Study of Web Traffic*. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp 889–892
33. Sun C et al (2021) *Te-esn: Time encoding echo state network for prediction based on irregularly sampled time series data.* arXiv preprint http://arxiv.org/2105.00412
34. Wu Y et al (2021) *Dynamic gaussian mixture based deep generative model for robust forecasting on sparse multivariate time series.* In: Proceedings of the AAAI Conference on Artificial Intelligence 35(1): 651–659
35. Li L et al (2021) *Learning interpretable deep state space model for probabilistic time series forecasting.* arXiv preprint http://arxiv.org/2102.00397
36. Bińkowski M et al (2018) *Autoregressive convolutional neural networks for asynchronous time series.* In: International Conference on Machine Learning, PMLR, pp 580–589
37. Chen Z et al (2021) *Time-Aware Multi-Scale RNNs for Time Series Modeling*. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, pp 2285–2291
38. Yang L et al (2020) *Html: Hierarchical transformer-based multi-task learning for volatility prediction.* In: Proceedings of The Web Conference 2020, pp 441–451
39. Yu R et al (2017) *Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting*. In: Proceedings of the 2017 SIAM international Conference on Data Mining, Society for Industrial and Applied Mathematics, pp 777–785
40. Elman JL (1990) Finding structure in Time. Cogn Sci 14(2):179–211
41. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

42. Cho K et al (2020) *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* arXiv preprint http://arxiv.org/abs/1406.1078
43. Lipton ZC, Berkowitz J, Elkan C (2015) *A Critical Review of Recurrent Neural Networks for Sequence Learning.* arXiv preprint http://arxiv.org/abs/1506.00019
44. Chung J et al (2014) *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.* arXiv preprint http://arxiv.org/abs/1412.3555
45. Chen G (2016) *A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation.* arXiv preprint http://arxiv.org/abs/1610.02583
46. Sherstinsky A (2020) *Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network*. Physica D: Nonlinear Phenomena 404:132306
47. Han K et al (2021). Transformer in transformer. Advances in neural information processing systems 34:15908–15919
48. Kitaev N et al (2020) *Reformer: The Efficient Transformer.* arXiv preprint http://arxiv.org/abs/2001.04451
49. Li S et al (2019) *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting.* Advances in neural information processing systems 32
50. Brown TB et al (2020) *Language Models are Few-Shot Learners.* arXiv preprint http://arxiv.org/abs/2005.14165
51. Shen L, Wang Y (2022) TCCT: tightly-coupled convolutional transformer on time series forecasting. Neurocomputing 480:131–145
52. Chen K et al (2021) *NAST: Non-Autoregressive Spatial-Temporal Transformer for Time Series Forecasting.* arXiv preprint http://arxiv.org/abs/2102.05624
53. Wu N et al (2020) *Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case.* arXiv preprint http://arxiv.org/abs/2001.08317
54. Lim B et al (2021) Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. Int J Forecast, 37(4):1748–1764
55. Cai L et al (2020). Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. Trans GIS, 24(3):736–755
56. Wu S et al (2020) *Adversarial Sparse Transformer for Time Series Forecasting.* Adv Neural Inf Process Syst 33:17105–17115
57. Qi X et al (2021) *From Known to Unknown: Knowledge-guided Transformer for Time-Series Sales Forecasting in Alibaba.* arXiv preprint http://arxiv.org/abs/2109.08381
58. Madhusudhanan K et al (2021) *Yformer: U-Net Inspired Transformer Architecture for Far Horizon Time Series Forecasting.* arXiv preprint http://arxiv.org/abs/2110.08255
59. Tipirneni S, Reddy CK (2021) *Self-supervised Transformer for Multivariate Clinical Time-Series with Missing Values.* arXiv preprint http://arxiv.org/abs/2107.14293
60. Xu J (2021) *Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy.* arXiv preprint http://arxiv.org/abs/2110.02642
61. Song H et al (2018) *Attend and diagnose: Clinical time series analysis using attention models.* In: Proceedings of the AAAI conference on artificial intelligence 32(1)
62. Liu M et al (2021) *Gated Transformer Networks for Multivariate Time Series Classification.* arXiv preprint http://arxiv.org/abs/2103.14438
63. Vaswani A et al (2017) Attention is All you Need. Advances in Neural Information Processing Systems.
64. Woo G et al (2022) Etsformer: Exponential smoothing transformers for time-series forecasting. arXiv preprint http://arxiv.org/abs/2202.01381
65. Tang B, Matteson DS (2021) Probabilistic transformer for time series analysis. Adv Neural Inf Process Syst 34:23592–23608
66. Cui Y, Xie J, Zheng K (2021) Historical inertia: a neglected but powerful baseline for long sequence time-series forecasting.In: Proceedings of the 30th ACM international conference on information & knowledge management, pp 2965–2969
67. Wu H et al (2021) *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. Advances in neural information processing systems 34:22419–22430
68. Zeng P et al (2022) Muformer: a long sequence time-series forecasting model based on modified multi-head attention. Knowl Based Syst 254:109584
69. Chang S et al (2017) *Dilated Recurrent Neural Networks.* Advances in neural information processing systems 30
70. Liu S et al (2022) *Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting.* In: The Tenth International Conference on Learning Representations
71. Song W, Fujimura S (2021) Capturing combination patterns of long- and short-term dependencies in multivariate time series forecasting. Neurocomputing 464:72–82

72. Hu J, Zheng W (2019) *Transformation-gated LSTM: efficient capture of short-term mutation dependencies for multivariate time series prediction tasks*. In: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–8
73. Huang S et al (2019) *DSANet: Dual Self-Attention Network for Multivariate Time Series Forecasting*. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 2129–2132
74. Zhao X et al (2022) *Generalizable Memory-driven Transformer for Multivariate Long Sequence Time-series Forecasting*. arXiv preprint http://arxiv.org/abs/2207.07827
75. Wang X et al (2022) Long Time Series Deep forecasting with Multiscale feature extraction and Seq2seq attention mechanism. Neural Process Lett 54(4):3443–3466
76. Liu Y et al (2022) *Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting.* Advances in Neural Information Processing Systems 35:9881–9893
77. Lin Y et al (2020) *SpringNet: Transformer and Spring DTW for Time Series Forecasting.* In: Neural Information Processing: 27th International Conference, Proceedings, Part III 27, pp 616–628
78. Lee WK (2020) *Partial Correlation-Based Attention for Multivariate Time Series Forecasting.* In: Proceedings of the AAAI Conference on Artificial Intelligence, 34(10):13720–13721
79. Zhou T et al (2022) *FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting.* In: International conference on machine learning, PMLR, pp 27268–27286
80. Chu J, Cao J, Chen Y (2022) An Ensemble Deep Learning Model Based on Transformers for Long Sequence Time-Series Forecasting. In: International Conference on Neural Computing for Advanced Applications. pp 273–286
81. Chen W et al (2022) Learning to Rotate: Quaternion Transformer for Complicated Periodical Time Series Forecasting. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, pp 146–156
82. Li Y et al (2023) Towards Long-Term Time-Series Forecasting: Feature, Pattern, and Distribution. In: IEEE 39th International Conference on Data Engineering (ICDE), pp 1611–1624
83. Yu F et al (2017) *Dilated Residual Networks.* In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 472–480
84. Gupta A, Rush AM (2017) *Dilated Convolutions for Modeling Long-Distance Genomic Dependencies.* arXiv preprint http://arxiv.org/abs/1710.01278
85. Ariyo AA et al (2014) *Stock Price Prediction Using the ARIMA Model.* In: 2014 UKSim-AMSS 16th international conference on computer modelling and simulation, IEEE, pp 106–112
86. Taylor et al (2018) *Forecasting at Scale.* Am Stat 72(1):37–45
87. Bahdanau D et al (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint http://arxiv.org/abs/1409.0473
88. Lai G et al (2018) Modeling long- and short-term temporal patterns with deep neural networks. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp 95–104
89. Flunkert V, Salinas D, Gasthaus J (2020) Deepar: probabilistic forecasting with autoregressive recurrent networks. Int J Forecasting, 36(3):1181–1191
90. Bracewell RN (1983) *The Fourier transform and its applications. 2nd ed., 3rd printing.*
91. Farge M (1992) Wavelet transform and their application to turbulence. Annu Rev Fluid Mech, 24:395–457.
92. Devlin J (2018) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv preprint http://arxiv.org/abs/1810.04805
93. Bache K, Lichman M (2013) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.
94. Tuli S et al (2022) *TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data.* arXiv preprint http://arxiv.org/abs/2201.07284
95. Lin Y, Koprinska I, Rana M (2021) *SSDNet: State Space Decomposition Neural Network for Time Series Forecasting.* In: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, pp 370–378
96. Wang X et al (2022) Variational transformer-based anomaly detection approach for multivariate time series. Measurement, 191:110791
97. Yu C et al (2020) *Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction*. In: Computer Vision–ECCV 2020: 16th European Conference, Proceedings, Part XII 16, pp 507–523
98. Xu M et al (2020) *Spatial-Temporal Transformer Networks for Traffic Flow Forecasting.* arXiv preprint http://arxiv.org/abs/2001.02908
99. Yang C et al (2021) Transformer embeddings of irregularly Spaced events and their participants. arXiv preprint http://arxiv.org/abs/2201.00044
100. Zeng A et al (2023) *Are transformers effective for time series forecasting?* In: Proceedings of the AAAI conference on artificial intelligence 37(9):11121–11128

101. Nie Y et al (2022) A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. arXiv preprint http://arxiv.org/abs/2211.14730
102. Dwivedi VP, Bresson X (2020) *A generalization of transformer networks to graphs.* arXiv preprint http://arxiv.org/abs/2012.09699
103. Rong Y et al (2020) Self-supervised graph transformer on large-scale molecular data. Adv Neural Inf Process Syst 33:12559–12571
104. Harvey AC et al (2007) Trends and cycles in economic time series: a bayesian approach. J Econ 140(2):618–649
105. Yuan X et al (2020) Integrated Long-Term Stock Selection models based on feature selection and machine learning algorithms for China Stock Market. IEEE Access 8:1–1
106. GeWenbo et al (2022) Neural network–based financial volatility forecasting: a systematic review. ACM Comput Surv (CSUR) 55(1):1–30
107. Yang B et al (2001) An early warning system for loan risk assessment using artificial neural networks. Knowl Based Syst 14(5–6):303–306
108. Livieris IE et al (2020) A CNN–LSTM model for gold price time-series forecasting. Neural Comput Appl 32(23):17351–17360
109. Hong T et al (2016) Probabilistic energy forecasting: global energy forecasting competition 2014 and beyond. Int J Forecast 32(3):896–913
110. Wu Z et al (2020) Connecting the dots: Multivariate Time Series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 753–763
111. Chang YY et al (2018) *A Memory-Network Based Solution for Multivariate Time-Series Forecasting.* arXiv preprint http://arxiv.org/abs/1809.02105
112. Demeniconi C, Davidson I (2021) Proceedings of the 2021 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics
113. Shen Z et al (2020) A novel time series forecasting model with deep learning. Neurocomputing 396:302–313
114. Yang Z et al (2022) Adaptive temporal-frequency network for time-series forecasting. IEEE Trans Knowl Data Eng 34(4):1576–1587
115. Hou X et al (2020) An enriched time-series forecasting Framework for Long-Short Portfolio Strategy. IEEE Access 8:31992–32002
116. Yoshimi S et al (2020) *Forecasting Corporate Financial Time Series using Multi-phase Attention Recurrent Neural Networks.* In: EDBT/ICDT Workshops
117. Zhao Y et al (2018) *Forecasting Wavelet Transformed Time Series with Attentive Neural Networks*. In: 2018 IEEE international conference on data mining (ICDM), IEEE, pp 1452–1457
118. Fu CW, Nguyen TT (2003) *Models for Long-Term Energy Forecasting*. In: 2003 IEEE Power Engineering Society General Meeting, IEEE, 1:235–239
119. Khuntia SR et al (2016) *Forecasting the load of electrical power systems in mid- and long-term horizons: a review.* IET Gener Transm Dis 10(16):3971–3977
120. Hecke TV (2012) Power study of anova versus Kruskal-Wallis test. J Stat Manage Syst 15(2–3):241–247
121. Yoo J, Kang U (2021) *Attention-Based Autoregression for Accurate and Efficient Multivariate Time Series Forecasting.* In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), Society for Industrial and Applied Mathematics, pp 531–539
122. Pang Y et al (2018) *Hierarchical Electricity Time Series Forecasting for Integrating Consumption Patterns Analysis and Aggregation Consistency*. In: Twenty-Seventh International Joint Conference on Artificial Intelligence, pp 3506–3512
123. Bogaerts T et al (2020) A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transp Res Part C: Emerg Technol 112:62–77
124. Qu L et al (2019) Daily long-term traffic flow forecasting based on a deep neural network. Expert Syst Appl 121:304–312
125. Chen Q et al (2016) *Learning deep representation from big and heterogeneous data for traffic accident inference*. In: Proceedings of the AAAI Conference on Artificial Intelligence 30(1)
126. Pan Z et al (2021) *AutoSTG: Neural Architecture Search for Predictions of Spatio-Temporal Graph.* In: Proceedings of the Web Conference, pp 1846–1855
127. Han L et al (2021) *Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting*. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp 547–555
128. Cirstea RG et al (2021) *EnhanceNet: Plugin Neural Networks for Enhancing Correlated Time Series Forecasting*. In: 37th International Conference on Data Engineering (ICDE), IEEE, pp 1739–1750
129. Elmi S, Tan K-L (2021) DeepFEC: Energy Consumption Prediction under Real-World Driving Conditions for Smart Cities. In: Proceedings of the Web Conference 2021. pp 1880–1890

130. Zhang J, Zheng Y, Qi D (2016) *Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction.* In: Proceedings of the AAAI conference on artificial intelligence, 31(1)
131. Liang Y et al (2021) *Fine-Grained Urban Flow Prediction*. In: Proceedings of the Web Conference, pp 1833–1845
132. Li Y, Moura JMF (2020) *Forecaster: A Graph Transformer for Forecasting Spatial and Time-Dependent Data*. In: European Conference on Artificial Intelligence (ECAI), pp 1293–1300
133. Dou K, Sun X (2021) *Long-Term Weather Prediction Based on GA-BP Neural Network.* In: IOP Conference Series: Earth and Environmental Science, 668(1):012015
134. Ward SN (1995) Area-based tests of long-term seismic hazard predictions. Bull Seismol Soc Am 85(5):1285–1298
135. Pandit R et al (2022) Sequential data-driven long-term weather forecasting models' performance comparison for improving offshore operation and maintenance operations. Energies, 15(19):7233
136. Qi Y et al (2019) *A hybrid model for spatiotemporal forecasting of PM2.5 based on graph convolutional neural network and long short-term memory.* Sci Total Environ, 664(MAY 10):1–10
137. Lauffenburger JC et al (2018) Predicting Adherence to Chronic Disease medications in patients with long-term initial medication fills using indicators of clinical events and Health behaviors. J Managed Care Specialty Pharm 24(5):469–477
138. Sanson G et al (2020) Prediction of early- and long-term mortality in adult patients acutely admitted to internal medicine: NRS-2002 and beyond. Clin Nutr 39(4):1092–1100
139. Zeroual A et al (2020) Deep learning methods for forecasting COVID-19 time-series data: a comparative study. Chaos, Solitons & Fractals 140:110121
140. Marling C, Bunescu R (2020) The OhioT1DM dataset for blood glucose level prediction: Update 2020. Inform Technol Nanatechnol 2675:71–74
141. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22(4):679–688
142. Armstrong J et al (2002) Principles of forecasting: a handbook for researchers and practitioners. Int J Forecast 18(3):468–478
143. Coleman CD, Swanson DA (2007) On MAPE-R as a measure of cross-sectional estimation and forecast accuracy. J Econ Soc Meas 32(4):219–233
144. Armstrong JS, Collopy F (1992) Error measures for generalizing about forecasting methods: empirical comparisons. Int J Forecast 8(1):69–80
145. Cleveland RB, Cleveland WS (1990) STL: a seasonal-trend decomposition procedure based on Loess. J Official Stat, 6:3–73
146. Hamilton JD (2020) *Time Series Analysis. Princeton University Press.*
147. Taieb SB, Hyndman RJ (2012) *Recursive and direct multi-step forecasting: the best of both worlds.* In: Proceedings of the Web Conference 2021, pp 1846–1855
148. Chevillon G (2007) Direct multi-step estimation and forecasting. J Economic Surveys 21(4):746–785