Transformer Architectures in Time Series Analysis: A Review

S. Thundiyil[1*], J. Picone[2] and S. McKenzie[3]

1. Dept. of Elect. and Comm. Eng., BMS Institute of Technology and Management, Bengaluru, IN
2. The Neural Engineering Data Consortium, Temple University, Philadelphia, Pennsylvania, USA
3. University of New Mexico Health Sciences, Albuquerque, New Mexico, USA

(saneesh@bmsit.in, picone@temple.edu, sammckenzie@salud.unm.edu)

Abstract

Analysis of time series data for classification or prediction tasks is very useful in various applications such as healthcare, climate studies and finance. As big data resources have recently become available in many fields, it is now possible to apply extremely high dimensional deep learning models that can model long-term temporal and spatial context. Traditional methods such as autoregressive integrated moving average (ARIMA), long short-term memory networks (LSTM), gated recurrent units (GRUs) and recurrent neural networks (RNN) have provided robust frameworks in the analysis of time series data. However, these methods have had limited success when applied to applications where long-term context is crucial. Transformer-based architectures such as ChatGPT have emerged as a powerful method for this class of problems. In this review, we provide a detailed analysis of state of the art in deep learning systems that model long-term context. We review eleven transformer-based architectures that have been successfully applied to healthcare-related applications involving time series or high resolution image data. We have focused on enhanced transformer architectures that can solve important challenges such as segmentation, forecasting, and classification.

1. Introduction

Time series data analysis involves the examination of datasets composed of time-ordered entries. This analysis is crucial in many fields for predicting future trends, understanding past behaviors, and making informed decisions. Time series data analysis is a fundamental aspect of statistical studies and data science, playing a critical role in numerous fields ranging from healthcare and finance to climate science and engineering. The core idea of time series analysis is to understand, model, and predict temporal data. The values in time series data are recorded at successive points in time, often at equally spaced intervals and hence the data is inherently sequential.

Time series data possesses several distinct characteristics that are critical for its analysis and interpretation. Some of the important characteristics of time series analysis are autocorrelation, trend and seasonality [1], [2], [3]. Autocorrelation in time series data refers to the degree of correlation between a time series and a lagged version of itself. It shows how much similar or related the data points are to their past values within the series:

$$R(\tau) = \frac{E[(x(t) - \mu)(x(t + \tau) - \mu)]}{\sigma^2}. \quad (1)$$

In Eq. 1 we show the autocorrelation $R(\tau)$ of a time series $x(t)$ at lag τ, where E is the expected value, μ is the mean of the time series, and σ^2 is the variance of the time series.

Trend refers to the long-term movement or direction in the data over time, disregarding short-term fluctuations. It represents the underlying tendency of the data to increase, decrease, or remain stable over a long period. Trends can be linear or nonlinear and can vary in slope and shape depending on the nature of the data and the factors influencing it.

Seasonality captures the regular fluctuations or patterns that occur at specific regular intervals, such as daily, weekly, monthly, or yearly. This is especially common in data related to weather, retail sales, and

* Corresponding Author: Saneesh Cleatus Thundiyil, Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Doddaballapura Main Road, Avalahalli, Bengaluru, Karnataka, India – 560064. Tel: +91 97 3138 2840. Fax: +91 80 6873 0462. Email: saneesh@bmsit.in.

energy consumption, where the time of year significantly influences the data. Sometimes such seasonal variations can be at irregular intervals. These irregular intervals are referred as cyclical variations rather than seasonality.

Stationarity is another characteristic of time series data where statistical properties such as mean, variance, and autocorrelation are constant over time. Many time series models assume that the data is stationary, or they involve transforming the data to achieve stationarity. The random variation in the data that cannot be attributed to trend, seasonality, or cycles may be considered as noise. Noise is inherently stochastic and unpredictable, often referred to as the "error" or "residual" part of a time series.

Careful analysis of time series data is crucial in many diverse domains since it enables predictive analytics and gives us insights into important temporal patterns. In the financial sector, time series data such as stock prices, exchange rates, and economic indicators like GDP and inflation rates are essential for market analysis and forecasting [4]. Financial time series are characterized by their volatility and are often analyzed for trend detection, anomaly identification, and risk assessment [5],[6]. Similarly, environmental, and climatological time series, including temperature recordings, rainfall measurements, and air quality indices, play a vital role in climate modeling and environmental research. These datasets are integral for understanding long-term climate patterns, seasonal variations, and environmental change assessments [7].

Biomedical time series data, including heart rate monitoring and EEG recordings, are fundamental in patient health monitoring and medical research [8]. In speech signal processing, time series analysis has been applied to enhance voice recognition systems and improve human-computer interaction. Algorithms for speech signal analysis have been developed to extract features in both time and frequency domains, providing valuable insights for speech recognition and processing [9]. In industrial settings, time series data such as production levels and equipment performance metrics aid in optimizing operations and predictive maintenance. Retail and business analytics heavily rely on time series data for sales forecasting and understanding consumer behavior patterns.

Large language model (LLM) based algorithms are used for generating auxiliary virtual images, demonstrating significant improvements in image processing tasks. The use of a transformer architecture to enrich feature diversity of images, showcases the potential of LLMs in image processing applications [10]. Furthermore, the integration of LLMs in image processing highlights the significance of spatial and temporal contexts. Spatial context often requires detailed analysis within a snapshot, while temporal context benefits from long-term models that track changes over time.

Each of these time series data categories, with their unique properties and patterns, require specialized analytical techniques. From stochastic models and machine learning algorithms to signal processing and statistical methods, the insights derived from these analyses are pivotal in decision-making processes across various sectors. For example, in Figure 1, we show the stock market variation in Dow Jones from Jan 2023 to Feb 2024. Trend is very important in such signals, and such signals are not zero mean or easily modeled by stable linear systems.

In Figure 2, we show a satellite image of glacier shrinkage due to climatic conditions. It is to be noted the time series data need not always be one dimensional as in stock market or biomedical signals. The idea of spatial context, the ability to model the relationship between adjacent pixels, is important in applications, such as image and video processing, environmental modeling, and geographical information systems. In such applications, the data encapsulates not only the change over time but also the intricate spatial interconnections between data points. For instance, in satellite imagery analysis used for environmental monitoring or urban planning, each pixel's value evolves over time, reflecting changes due to natural events, human activities, or seasonal cycles.

In Figure 3, we provide another example of time series data obtained from the EEG recordings. Such signals, which are multichannel in nature, have both temporal and spatial dependencies. Here, spatial dependencies mean correlations between channels, where each channel corresponds to a signal collected from a sensor placed in a specific location on the scalp. A temporal event, such as a seizure, occurs on multiple channels which are physically close to one another.

2. Traditional Time Series Analysis Techniques

Traditional methods for time series analysis have evolved over the years. To analyze time series data, classical approaches such as autoregressive models were widely used, especially for prediction tasks. These models predict future data points using a linear combination of past values. The autoregressive model assumes that the output variable depends linearly on its previous values and a stochastic term. Talwar [11] explored various autoregressive models for dynamic forecasting of equity markets, emphasizing the use of past data in predicting future volatility. Bondon [12] provided an explicit formula for the prediction error of future values of a stationary process with incomplete past data, highlighting the use of autoregressive processes. Madadi et al. [13] expanded autoregressive models to forecast dynamic line ratings in power networks, addressing the trend and fluctuation of past data. Ray [14] discussed the use of mid-prediction filters in autoregressive models for separating the nonstationary part of a signal. Hall et al. [15] explored high-dimensional generalized linear autoregressive models, offering insights into predicting future observations using current and past observations. Engle [16] introduced autoregressive conditional heteroscedastic (ARCH) processes, a class of stochastic processes used for forecasting with nonconstant variances conditional on the past. Rather [17] presented an autoregressive neural network approach for predicting stock returns, emphasizing the use of past values in regression variables.

2.1. Correlation-Based Methods

Moving average (MA) models use the past forecast errors in a regression-like model. A moving average model helps in smoothing out noise from random fluctuations in time series data. Autoregressive moving average (ARMA) models combine both AR and MA models to describe time series data using both autoregressive terms and moving average terms. Loneck & Zurbenko [18] discussed the Kolmogorov-Zurbenko periodogram with DiRienzo-Zurbenko smoothing for spectral analysis of time series data, comparing its performance to traditional ARIMA algorithms. Sun et al. [19] proposed an MA method based on complex exponential decomposition for noise elimination in non-stationary and non-linear signals.

An extension of ARMA, autoregressive integrated moving average models (ARIMA) includes differencing of raw observations (integration) to make the time series stationary, which is a common requirement for AR and MA models. Lee et al. [20] applied an ARIMA model to predict future network throughput, crucial for improving network protocols. Garg et al. [21] used the ARIMA model to analyze long-term noise monitoring data in traffic noise pollution studies. Valipour et al. [22] estimated the ability of ARIMA models in forecasting the monthly inflow of Dez dam reservoir. Sameh & El-shabrawy [23] investigated the application of ARIMA and SARIMAX models in the context of climate change time series forecasting. Pitfield [24] compared the efficiency of ARIMA and regression models in simulating air-transport passengers by route.

The Box-Jenkins methodology [1],[35] is a systematic method for applying an autoregressive integrated moving average (ARIMA) model. Haviluddin & Alfred [36] presented an approach for network traffic characterization using the ARIMA technique, demonstrating its application in modeling internet network traffic. Jafarian-Namin et al. [37] focused on modeling and forecasting the yearly inflation rate of Iran using ARIMA, employing the Box-Jenkins methodology to confirm the effectiveness of different ARIMA models. Duarte et al. [38] compared Box and Jenkins methodologies with Artificial Neural Networks in time series forecasting, highlighting the performance of ARIMA and Transfer Function Models versus Neural Network Models. Jamii et al. [39] aimed to predict carbon dioxide emissions in Morocco using the Box-Jenkins ARIMA approach, demonstrating the application of this methodology in environmental modeling.

Seasonal decomposition techniques decompose a time series into seasonal, trend, and residual components, typically using moving averages. Dozie & Ibebuogu [25] discussed the decomposition with the mixed model using the Buys-Ballot method, emphasizing the estimation of trend parameters, seasonal indices, and residual components. Hebbel & Heiler [26] presented a method for decomposing a time series into trend-cyclical and seasonal components, using a smoothness criterion and goodness of fit criterion. He et al. [27] developed a seasonal-trend decomposition-based dendritic neuron model (STLDNM) for financial time series prediction, highlighting the effectiveness of seasonal-trend decomposition in complex data series. Sulandari et al. [28] combined deterministic function and neural network models to forecast time series with trend and seasonal patterns, utilizing singular spectrum analysis (SSA) for decomposition. Lacroix [29] explored short-term analysis and business cycle estimation using seasonal decomposition, focusing on the consistency of methodologies in seasonal adjustment and trend-cycle estimation. Zhang & Li [30] proposed a novel decomposition and combination technique for forecasting electricity consumption, using STL decomposition to separate trend, season, and residual components of time series.

Cross-correlation and autocorrelation analysis measure the relationship between the time series and lagged versions of itself or another time series. Dean & Dunsmuir [51] highlight the dangers of cross-correlation in time series analysis within various fields, emphasizing the importance of constructing transfer function autoregressive models to avoid spurious relationships due to autocorrelation. Olden & Neff [52] discuss the biases in cross-correlation analysis caused by intra-multiplicity (the time lags observed and the cross-correlation coefficients that are computed within a pair of time series) even in the absence of autocorrelation, and provide formulas to quantify and minimize these biases. Taylor [53] explains how autocorrelations, correlograms, and plots of the autocorrelation function can reveal the structure of a cycle within time-series data, providing statistical methods for deeper analysis. Zhang, Huang, Shekhar, & Kumar [54] utilize spatial autocorrelation to propose new processing strategies for correlation-based similarity range queries and joins, offering a novel approach to managing the computational cost of correlation analysis in spatial time series datasets. Stattegger [55] employs time series analysis techniques like autocorrelation and cross-correlation to reconstruct tectonic structures from geochemical drill hole log data, demonstrating the application of these methods in geology.

2.2. Frequency Domain and Multi-Timescale Based Methods

Fourier analysis techniques transform a time series into its frequency components. This is particularly useful in signal processing and in situations where periodic patterns need to be identified. Kaiser [31] discussed windowed Fourier transforms, highlighting their utility in providing information about signals simultaneously in the time and frequency domains, which is essential in signal processing. Bradford [32] examined time-frequency analysis methods, including Fourier Transforms, for analyzing systems with changing dynamic properties, underlining their importance in civil engineering and seismology. Kolawole [33] covered frequency analysis of signals using Fourier series and Fourier transform, emphasizing its role in signal processing and systems design. Vergura et al. [34] conducted a time-frequency analysis using Fourier and Wavelet transforms to detect properties of power required by different types of users, showcasing the application of Fourier analysis in power systems.

Spectral analysis uses the frequency spectrum contained in time series data, which is particularly useful in fields like seismology and electrophysiology. Ghaderpour et al. [56] introduced the antileakage least-squares spectral analysis for seismic data regularization and random noise attenuation, offering solutions to the spectral leakage problem. Baisch & Bokelmann [57] presented a method for spectral analysis of non-equidistantly spaced time series, applying the CLEAN algorithm to seismological data to detect temporal changes in elastic wave velocities. Ghil et al. [7] reviewed advanced spectral methods for climatic time series, illustrating connections between time series analysis and nonlinear dynamics, and discussing signal-to-noise enhancement.

Wavelet analysis breaks down time series data into different frequency components and studying each component with a resolution matched to its scale. Karim et al. [64] explored the use of wavelets (symlet 16) to detect business cycles in Malaysia by decomposing time series to study long-run trends and high-frequency components. Bartosch & Wassermann [65] presented a wavelet coherence method to display local coherence information between two seismic stations, applying it to seismic near-field data from Stromboli volcano. Masuda & Okabe [66] discussed the application of the wavelet transform for stationarity analysis and predictions in time series, allowing the observation of series in both time and frequency domains simultaneously. Schiff [67] adapted a noise reduction technique for time-series data using wavelets, presenting a method that filters noise using control surrogate data sets. Torrence & Compo [68] provided a practical guide to wavelet analysis with examples from the El Niño–Southern Oscillation (ENSO), including statistical significance tests for wavelet power spectra.

Exponential smoothing techniques include methods like Simple Exponential Smoothing for univariate data without trend or seasonality, and Holt-Winters' Exponential Smoothing for data with trend and/or seasonality. Gelper et al. [40] presented robust versions of exponential and Holt-Winters smoothing methods suitable for forecasting univariate time series in the presence of outliers, offering a recursive updating scheme for pre-cleaned data. Taylor & McSharry [41] evaluated univariate forecasting methods using European electricity demand data, highlighting the performance of double seasonal Holt-Winters exponential smoothing among other methods for predicting up to a day-ahead demand. Luoman [42] introduced three kinds of exponential smoothing — Simple, Holt and Winters. These are applicable to time series data with a variety of characteristics including trend and seasonality.

2.3. Nonlinear Methods

There were methods developed to deal with time series data exhibiting nonlinear behaviors, which cannot be captured adequately by linear models. Hegger et al. [58] describe the TISEAN package, which implements methods of nonlinear time series analysis based on deterministic chaos, covering data representation, prediction, noise reduction, dimension and Lyapunov estimation, and nonlinearity testing. Small [59] focuses on time series embedding and reconstruction, essential for analyzing experimental time series data with nonlinear methods, including discussions on determinism and stationarity in physiological data. Bradley & Kantz [60] revisit nonlinear time-series analysis, discussing the practical issues that restrict the approach's power, such as signal sampling and noise, and highlighting its successful application across thousands of real and synthetic data sets.

Kantz [61] discusses the potentials and limitations of nonlinear time series analysis, emphasizing the need for extensions of methods towards systems coupled to random noises and those with more than a few active degrees of freedom. Zou et al. [62] provide an in-depth review of complex network methods for characterizing dynamical systems based on time series, covering phase space-based recurrence networks, visibility graphs, and Markov chain-based transition networks. Pereda et al. [63] describe nonlinear multivariate analysis methods used in neurophysiology to study the relationship between simultaneously recorded signals, covering concepts of phase synchronization, generalized synchronization, and event synchronization.

2.4. Regression-Based Methods

Identifying and analyzing trends in time series data often requires statistical techniques to model and forecast future values based on observed trends. Neves & Cordeiro [43] presented an approach integrating exponential smoothing and bootstrap methodologies for time series prediction, emphasizing the importance of selecting the best model for accurate forecasts. Zavala & Messina [44] provided a statistical framework based on dynamic harmonic regression for examining modal behavior, trend extraction, and forecasting in wind power generation, showcasing the flexibility of time series models. Miah [45] explored techniques for the analysis of financial data using time series models, demonstrating how to analyze and forecast economic indicators and perform trend analysis.

Jha et al. [46] investigated contemporary approaches for forecasting vehicular population in India, comparing trend line analysis, econometric analysis, and time series (TS) analysis, and found TS analysis to be more accurate. Wonu & Orlu [47] modeled time-series data on senior secondary student mathematics achievement over 29 years, using trend analysis and ARIMA techniques to forecast future values, highlighting the effectiveness of these methods in educational data analysis. Idrees et al. [48] discussed analyzing the Indian stock market using time series data to build a statistical model for efficient future stock predictions. This research demonstrates the significance of time series analysis in financial markets for uncovering market trends and forecasting stock performance. Rivera [49] emphasized the role of stationarity in business and economic research, discussing the importance of identifying non-stationary time series and the need for stationarity in the data prior to analysis. Hu [50] introduced the combination of time series forecasting with topological data analysis as a technique to solve real-world problems, using COVID-19 pandemic data as a case study.

In this section we have discussed the traditional time series analysis methods ranging from auto-regressive models, which leverage past values for predictions, and moving average models, aimed at smoothing out noise, to more complex Autoregressive Integrated Moving Average (ARIMA). These techniques have been applied across various domains, such as finance, climate studies, biomedical, speech signal etc. Techniques such as seasonal decomposition and Fourier analysis are used to identify the periodic patterns whereas exponential smoothing and trend analysis provide tools for handling data with or without seasonal variations. Spectral, wavelet, and nonlinear time series analyses offer advanced methods for dealing with complex data structures. The variety of methodologies discussed in this section highlights the evolution of time series analysis in capturing and forecasting the intricate behaviors of sequential data across various fields.

3. Modern Approaches in Time Series Analysis

Modern methods for time series analysis have significantly evolved, incorporating advanced statistical techniques, machine learning algorithms, and artificial intelligence. These methods are capable of handling large volumes of data, complex patterns, and non-linear relationships, making them suitable for a wide range of applications, from finance and business forecasting to environmental monitoring and healthcare. In this section, we highlight several approaches that represented fundamental advances in the field, or introduced paradigms that became the foundation for more advanced approaches.

3.1. Reinforcement Learning

Reinforcement Learning optimizes a cumulative reward metric to make decisions over time. Ansari et al. [69] proposed a novel decision support system for automated stock trading based on deep reinforcement learning, observing both past and future trends of stock prices to make optimal trading decisions. This study demonstrated the effective use of reinforcement learning in algorithmic trading and stock market forecasting. Aboussalah et al. [70] explored the value of the cross-sectional approach to deep reinforcement learning in dynamic asset allocation. This research provides insights into the effectiveness of reinforcement learning algorithms in financial applications, particularly in portfolio management.

Roy et al. [71] presented an augmented AI algorithmic trading approach that combines a Thick Data Heuristic with Deep Reinforcement Learning for day and swing trading order timing executions. The study shows the integration of AI and heuristics with deep learning techniques for effective trading decisions. Lei et al. (2020) proposed a time-driven feature-aware jointly deep reinforcement learning model (TFJ-DRL) for algorithmic trading, integrating deep learning with reinforcement learning for improved financial signal representation and decision-making [72]. Li et al. [73] introduced a robust deep reinforcement learning-based trading agent for algorithmic trading in dynamic financial markets, using deep Q-network and asynchronous advantage actor-critic for adapting to trading markets. Chen et al. [74] proposed an agent-based reinforcement learning system to mimic professional trading strategies, demonstrating its ability to reproduce trading decisions and improve convergence in dynamic environments.

3.2. Nonparametric Methods

Techniques such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVMs), and similar clustering algorithms are widely used for time series clustering and classification tasks. These methods are robust and powerful, and often are used to establish baseline performance for new data sets and applications. Chandralekha & Shenbagavadivu [75] explored clustering and classification in machine learning, particularly for predicting heart disease by analyzing various medical diagnostic parameters and patterns. They compared unsupervised learning (like K-means, K-modes, K-medoids) and supervised learning (such as SVM, Random Forest, Decision Tree, and k-NN). Senthil & Suseendran [76] proposed a Sliding Window Technique-based Improved Association Rule Mining with Enhanced SVM (SWT-IARM with ESVM) for time series data classification. This approach focuses on efficient rule discovery and classification by combining ESVM classification with IARM for more accurate rule classification.

Ougiaroglou et al. [77] explored the application of data reduction techniques as a preprocessing step before training Neural Networks and SVMs for time series classification. They also proposed a new data reduction technique based on the k-median clustering algorithm. Yang et al. [78] developed a kernel fuzzy c-means clustering-based fuzzy SVM algorithm (KFCM-FSVM) for dealing with classification problems involving outliers or noises, using FCM clustering in the high-dimensional feature space. Sathyamoorthy & Sivasankar [79] presented a hybrid approach where clustering algorithms are used to reduce the training dataset size, followed by applying complex algorithms like SVM and MLP for classification on the reduced data set.

Advanced algorithms such as Isolation Forest, One-Class SVM, and Autoencoders are used to identify unusual patterns or outliers in time series data, crucial in fraud detection and system health monitoring. Aguilar et al. [80] proposed the first interpretable autoencoder based on decision trees, designed to handle categorical data without the need to transform data representation. This model provides a natural explanation for experts in application areas and is among the top-ranked anomaly detection algorithms, along with One-class SVM and Gaussian Mixture. Park et al. [81] proposed multi-modal anomaly detection in embedded systems using time-correlated measurements of power consumption and memory accesses. They trained one-class SVM and isolation forest classifiers for anomaly detection, showing accurate detection of anomalies.

Ma & Perkins [82] introduced a new algorithm for time-series novelty detection based on one-class SVMs. They converted time-series into vectors in phase spaces and interpreted novel events as outliers of the “normal” distribution, showing promising performance of the algorithm. Alfeo et al. [83] proposed an anomaly detection approach based on deep learning for smart manufacturing. They combined an autoencoder with a discriminator based on general heuristics, proving the convenience of this approach against isolation forest in industrial applications. Yang et al. [84] proposed a high-dimensional anomaly detection algorithm based on isolated forest with a deep autoencoder (AE-IForest), mapping high-dimensional data to a low-dimensional space and fusing reconstruction error with data isolation scores for anomaly detection.

Derbentsev et al. [127] discuss short-term forecasting of cryptocurrency time series using random forests and a stochastic gradient boosting machine, highlighting the applicability of machine learning ensembles for forecasting cryptocurrency prices. Pop et al. [128] analyze the performance of random forests and gradient boosting algorithms in forecasting energy consumption, and compare them to a Weighted Average Ensemble Method. Mienye et al. [129] present a concise overview of ensemble learning, covering bagging, boosting, and stacking, and focuses on widely used ensemble algorithms, including random forest and gradient boosting.

3.3. Neural Networks

Convolutional Neural Networks (CNNs), primarily known for image processing, have been adapted for time series analysis. They can capture spatial-temporal patterns in data, making them useful for series with spatial components (like EEG signals). Liu et al. [100] proposed a multivariate convolutional neural network (MVCNN) for multivariate time series classification, integrating a tensor scheme with a novel deep learning architecture. Nakamura et al. [101] discussed using one-dimensional convolutional neural networks (1D-CNNs) for time series analysis with a method to mitigate noise interference by injecting noise into the data for feature extraction. Younis et al. [102] proposed a new approach to interpret CNN outputs for multivariate time series data by extracting and clustering activated time series sequences learned from a trained network. Chadha et al. [103] proposed permutation layers in CNNs to overcome inefficiencies in capturing features from unsorted "2D-images" formed by multivariate time-series analysis. Chervyakov et al. [104] focused on reducing the hardware cost of CNNs in applications like time series analysis, suggesting a CNN architecture based on the Residue Number System (RNS). Utama et al. [105] optimized CNN architecture for multivariate time-series data analysis using Particle Swarm Optimization (PSO), showing improvement in performance compared to ordinary CNNs.

Long Short Term Memory Network (LSTM) is a type of recurrent neural network (RNN) effective in complex time series forecasting due to its ability to model long-term dependencies. Manaswi [85] discusses the concepts of recurrent neural networks (RNNs) and LSTMs, highlighting their use in sequence prediction and time-series forecasting. Wu et al. [86] propose a new forecasting framework with LSTM models for forecasting Bitcoin daily prices, validating the excellent forecasting accuracy of the proposed models. Luo & Wang [87] introduce a long-term prediction model for time series using fuzzy information granules and recurrent fuzzy neural networks, integrating type-2 fuzzy sets and LSTM to improve anti-noise and memory ability. Kim et al. [88] propose a novel neural network architecture using a combination of LSTM and convolutional layers to predict time-series energy data with higher accuracy. Chen & Xu [89] developed a piecewise time series prediction model combining stacked LSTM network with a genetic algorithm, demonstrating its effectiveness in automatically selecting the proper structure according to the data.

Similar to LSTM, Gated Recurrent Units (GRU) are a type of RNN that are efficient in modeling temporal sequences and their long-range dependencies. They are used in situations where LSTMs might be too computationally intensive. Onyekpe et al. [90] proposed a Quaternion Gated Recurrent Unit (QGRU) for sensor fusion, leveraging quaternion algebra to map correlations within multidimensional features more efficiently than traditional GRUs. Tallec & Ollivier (2018) proved that learnable gates in recurrent models provide quasi-invariance to general time transformations in input data, leading to a new way of initializing gate biases in LSTMs and GRUs. Shen et al. [91] explored the use of GRU networks for predicting trading signals for stock indexes, comparing GRU-based models with traditional deep nets and support vector machines (SVM) [92]. Zheng & Chen [93] proposed a novel GRU model with selective state updating and adaptive mixed gradient optimization for accurate power time-series prediction.

Erichson et al. [94] introduced a novel gated recurrent unit with a weighted time-delay feedback mechanism to improve the modeling of long-term dependencies in sequential data. Dangovski et al. [95] developed the Rotational Unit of Memory (RUM), a phase-coded representation of the memory state in RNNs, which unifies unitary learning and associative memory, showing improved performance over LSTMs/GRUs. Morchid [96] proposed the Parsimonious Memory Unit (PMU) based on the assumption that short and long-term dependencies are related, showing better efficiency and processing time compared to GRU. Bilkhu et al. [97] implemented a Transformer-based model for video captioning using GRUs, showing improved performance on video captioning tasks. Hong et al. [98] proposed the Long Memory Gated Recurrent Unit (LMGRU) based on LSTM and GRU models, achieving better effectiveness and efficiency in time series classification tasks. Som et al. [99] utilized GRUs in combination with RNN for text classification, achieving a classification accuracy of 87% on a movie review dataset.

3.4. Deep Neural Networks

DeepAR is a probabilistic forecasting model with autoregressive recurrent networks. DeepAR provides accurate forecasting, especially for large datasets with many related time series. Jiang et al. [106] proposed an optimized DeepAR model using the Sparrow Search Algorithm for atmospheric PM2.5

prediction, demonstrating improved forecasting accuracy in both interval and point predictions. Dong et al. [107] introduced a real-time wireless monitoring system and employed the DeepAR model for deformation prediction of unstable slopes, showing good safety control ability and prediction accuracy. Jeon & Seong [108] modified the DeepAR model to address the intermittent and irregular characteristics of sales demand, achieving robust and stable predictions in time series forecasting. Consoli et al. [71] used economic news within a DeepAR framework to forecast the Italian 10-year interest rate spread, showing that a deep learning network outperforms classical methods. Park et al. [109] investigated DeepAR models for probabilistic forecasting of photovoltaic generations, evaluating the tightness of the prediction interval with normalized residues.

Shen et al. [110] proposed DeepARMA, an LSTM-based model derived from DeepAR, addressing weaknesses in rolling window size determination and noise neglect. Li et al. [111] built a model based on deep neural networks, combining convolutional and recurrent networks for multivariate time series analysis, including an autoregressive network. Gouttes et al. [112] proposed a method for probabilistic time series forecasting, combining an autoregressive recurrent neural network with Implicit Quantile Networks.

Prophet, developed by Facebook, is designed for forecasting with daily observations that display patterns on different time scales. It is particularly effective for handling outliers, missing data, and seasonal effects. Chuwang & Chen [113] employed the Box–Jenkins time series with the Facebook Prophet algorithm for forecasting daily and weekly passenger demand for urban rail transit stations, demonstrating better computational forecasting performance accuracy. Toharudin et al. [114] employed LSTM and Facebook Prophet models in air temperature forecasting, highlighting the performance of Prophet in managing complex data series.

Saiktishna et al. [115] analyzed stock market trends using FB Prophet, noting its improved performance and accuracy in prediction. Huang [116] utilized Facebook Prophet with macroeconomic regressors for forecasting stock prices, demonstrating its superiority in prediction accuracy compared to other models. Mahmud [117] predicted and analyzed COVID-19 daily cases in Bangladesh using the Facebook Prophet Model, demonstrating its capability in handling complex data series. Mphale et al. [118] proposed the Facebook-Prophet model for forecasting COVID-19 mortality in the SADC region, highlighting its effectiveness in prediction. Suresh et al. [119] conducted historical analysis and forecasting of the stock market using the FB Prophet model, emphasizing its improved performance in forecasting.

Vector Autoregression (VAR) is an extension of the AR model that captures the linear interdependencies among multiple time series. VAR models are widely used in econometrics. Lu [120] discusses the application of VAR in analyzing the dynamics among geographic processes and for spatial autoregressive modeling, providing an example of US population dynamics between 1910 and 1990. Myers et al. [121] use VAR methods to analyze the contribution of supply, demand, and policy shocks to fluctuations in the Australian wool market, comparing VAR procedures with conventional models. Alvarez-De-Toledo et al. [122] offer an approximation between econometric techniques and system dynamics methodology, showing how to simulate an SVAR model. McCracken et al. [123] assess forecasts from a mixed-frequency VAR to obtain intra-quarter forecasts of output growth as new information becomes available. Kilian & Lütkepohl [124] review the structural VAR approach in econometrics, contrasting it with other methodologies and highlighting its application in macroeconomics and finance.

3.5. Hybrid Methods

Ensemble Methods combine predictions from multiple models to improve forecasting accuracy. Methods like random forests, gradient boosting, and bagging are used in an ensemble manner for time series predictions. Galicia et al. [125]. This study presents ensemble models for forecasting big data time series, combining decision tree, gradient boosted trees, and random forest methods. The performance is evaluated on electricity consumption data, showing that the ensemble models outperform individual members. Valatsos et al. [126] predict critical time intervals for freight transportation using ensemble learning techniques, including bagging, random forest, and gradient boosting.

Levy & O'Malley [130] combined traditional statistical models with modern machine learning techniques to capture both linear and non-linear aspects of data. They introduced "Interaction Transformer", an algorithm that boosts logistic regression by integrating machine learning to identify interaction features from a random forest model. Chen [131] reviews models for predicting business bankruptcies, noting the shift from traditional statistical methodologies to machine learning techniques. The author emphasizes the role of hybrid classifiers, combining machine learning algorithms like support vector machines, decision trees, and genetic algorithms to improve the accuracy of bankruptcy prediction models. Anifowose et al. [132] present a hybrid machine learning approach to predict the formation cementation factor, combining the nonlinear feature selection capability of functional networks (FNs) with traditional artificial neural networks (ANNs). The FN-ANN hybrid model demonstrates improved accuracy and computational efficiency.

Von Rueden et al. [133] describe the combination of machine learning and simulation towards a hybrid modeling approach, suitable for applications based on both causal relationships and hidden dependencies represented in data. The authors discuss various types of combinations using simulation-assisted machine learning and machine learning-assisted simulation. Sadat et al. [134] developed a hybrid cryptographic framework for secure and efficient regression analysis over distributed data, combining somewhat homomorphic encryption and Intel Software Guard Extensions (Intel SGX). The framework ensures privacy while maintaining computational efficiency. These modern methods are often more flexible and powerful than traditional approaches, particularly in handling non-linear patterns, large datasets, and real-time analysis. They require a good understanding of the underlying models and appropriate preprocessing of data. The choice of method often depends on the specific characteristics of the time series data and the objectives of the analysis.

In Table 1, we provide a comparison of traditional methods for time series modeling and discuss the pros and cons of each approach. In Table 2, we provide a similar summary for modern approaches.

4. Long-Term Dependencies in Time Series Data

The temporal dependencies in time series data are crucial in various applications such as stock market prediction and fault diagnosis. These dependencies can span timeframes of a few hours to a few years making the analysis and classification of such data a challenging task. Time series data in energy systems, like wind turbines, inherently contain extremely long-term dependencies that are essential for forming classifiable features and effective fault diagnosis [135]. Biomedical time series data, such as EEG and ECG, do exhibit long-term dependencies, as demonstrated by Maiorana [136] in their study on the longitudinal behavior of EEG signals. This was further supported by Nakano [137], who found a relationship between the slowing of EEG and mental function decline in the elderly. The importance of capturing these long-term dependencies in predicting clinical events was highlighted by Li [138], who developed a hierarchical Transformer-based model for accurate prediction using longitudinal electronic health records. Zhao [139] also emphasized the need to retain sequential information in temporal data, which is crucial for prediction tasks in the biomedical domain.

The studies by Thombs [140], Lutz [141], Kim [142], and Jackson [143] collectively suggest that time series data from climate studies does exhibit long-term dependencies. Thombs and Kim both highlight the importance of analyzing historical time series data and the need for alternative adjustment methods to account for seasonality and long-term trends. Lutz and Jackson further emphasize the significance of longitudinal data in understanding the impact of climate change on forest ecosystems and ecological processes. These studies collectively underscore the presence of long-term dependencies in climate-related time series data.

Time series data obtained from financial analysis, such as stock market and inflation data, often exhibit long-term dependencies. This is due to the inherent nature of these data, which are characterized by sequential observations over time. These dependencies can be attributed to various factors, including the presence of heterogeneity, omitted variable bias, and duration dependence [144]. In the context of stock trading markets, univariate time series models have been found to be effective in certain cases, particularly in segments with sufficient historical data [145]. However, the effectiveness of these models may not be generalizable to all domains, particularly in forecasting after turning points. The presence of serial dependency in time series data can pose challenges in analysis, particularly when conventional methods that ignore this dependency are used [146]. Despite these challenges, time series analysis remains a valuable tool for understanding the underlying processes and patterns of change in financial data [147].

Despite these advancements, capturing long term dependencies and rare event detections is challenging. Modeling long-term dependencies poses what amounts to a combinatorial problem. Until the introduction of the so-called Large Language Model (LLM), this was an elusive problem. The Transformer model, which is based on an architecture that implements what is known as self-attention, has been a disruptive force in machine learning.

4.1. Introduction to the Transformer Architecture

The Transformer architecture (Figure 4), introduced by Vaswani et al. [148], leverages self-attention (scaled dot-product attention) as its core mechanism. This enables the model to assign importance weights to different parts of the input sequence, unlike recurrent and convolutional layers. These weights allow the Transformer to focus on relevant elements during processing, capturing long-range dependencies effectively. Central to self-attention is the computation of attention weights, which determine which parts of the input sequence are most relevant for a particular element. This eliminates the need for recurrent layers, which struggle with modeling long-range dependencies. In the original Transformer architecture, the input words or phrases are represented as vectors of real numbers in a high-dimensional space. This process is called input embedding and during this process the information about the order of the input sequence will be lost. Hence the authors introduced the concept of positional encoding which generates a vector informing the model about element positions within the sequence.

In Scaled Dot-Product Attention, the attention weights are computed as a function of the query (Q) and the key (K) matrices, scaled by the dimension of the keys (d_k):

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

where Q is the matrix of queries, K is the matrix of keys, V is the matrix of values, and d_k is the dimensionality of the key vectors.

A Transformer architecture enhances the ability of the model to focus on different positions by employing multiple heads for the attention mechanism. Each head captures different aspects of the attention. The output of each head is concatenated and linearly transformed into the desired dimensionality:

$$Multihead(Q, K, V) = \text{Concat}(head_1, head_2, \ldots\ldots, head_h)W^O \quad (3)$$

$$head_i = Attention\left(QW_i^Q, KW_i^H, VW_i^V\right) \quad (4)$$

where W_i^Q, W_i^K, W_i^V are the parameter matrices for i^{th} head, and W^O is the output linear transformation matrix.

In the original Transformer model, which is designed for natural language processing, positional encodings are added to the input embeddings to give the model information about the position of each word in a sentence. This concept is crucial for time series analysis as well, where the order of data points significantly impacts their meaning. For time series, positional encodings can be adapted to represent the sequential nature of the data more accurately, ensuring the model recognizes the temporal order of observations. This involves encoding not just the position within a sequence but also the actual time intervals between observations, which can be particularly important in irregularly sampled time series. Adjustments to the Transformer architecture, such as customizing the input and output layers or integrating domain-specific features, can help the model better interpret and predict these continuous values. By introducing mechanisms such as cyclic positional encodings into the model, a Transformer can recognize and predict these cyclic patterns more effectively. Researchers have introduced various modifications in the Transformer architecture, as shown in Figure 5, which enumerates application areas and domain specific architectures.

This review emphasizes applications in signal processing that addresses time series related tasks such as forecasting, classification, and anomaly detection. The popular architectures used for such applications include LogTrans [149], InParformer [150], Informer [151], Sageformer [152], Autoformer [153], Pyraformer [154], W-Transformers [155], FEDformer [156], Crossformer [157], Temporal Fusion Transformers (TFT) [158], [159], [160], and Transformer-XL [161].

4.2. LogTrans

The LogTrans architecture introduces an architecture that provides a combination of Transformer architecture and CNN parallel network for biomedical image segmentation [149]. CNNs excel at learning local dependencies within images. However, they tend to lack a broader understanding of the overall structure and relationships between different regions and components. LogTrans offers a hybrid approach using parallel branches consisting of a CNN and a transformer. The CNN branch focuses on extracting localized features (textures, edges, specific cell patterns), whereas the transformer branch specializes in learning global spatial relationships and contextual information.

The Separate-Combiner (SeCo) module is the heart of the LogTrans architecture. Instead of just jamming outputs from the two branches together, this module does two things: (1) separate – allows CNN and transformer features to further refine on their own, emphasizing relevant patterns for their specific focus; and (2) combiner: strategically fuses the refined features, enriching the representation. This gives the resulting segmentation the best of both worlds.

The LogTrans framework was evaluated on several biomedical datasets, including ablation studies on ISIC-2017 and UITNS-2022 as shown in Table 3.

4.3. Temporal Fusion Transformer

The Temporal Fusion Transformer (TFT) [139], [158], [160], [162], shown in Figure 7, integrates several components to handle different types of data and temporal relationships effectively. The core components include Gated Residual Networks, Variable Selection Networks, LSTM encoders, Multi-Head Attention, and Quantile forecasts. This architecture allows TFT to capture complex temporal patterns, handle missing data, and provide uncertainty estimates for forecasts. It is particularly effective in multi-horizon forecasting tasks, where predictions are needed over multiple future time steps.

B. Lim et al. [158] introduces an attention-based architecture for multi-horizon forecasting that combines high performance with interpretable insights into temporal dynamics. TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies. The architecture includes specialized components to select relevant features and gating layers to suppress unnecessary components, enabling high performance in a wide range of scenarios. The architectural innovations include gating mechanisms that allow the model to adaptively manage its depth and complexity, enabling efficient information processing across different scenarios without overfitting to less relevant data components.

The variable selection networks play a crucial role in identifying and focusing on the most relevant input variables for each forecasting step, thereby enhancing the model's accuracy and interpretability. TFT integrates information from static metadata using separate gated residual network (GRN) encoders to produce four different context vectors that are wired into various locations in temporal fusion decoder. TFT integrates vital background information into the forecasting process, allowing the model to condition its temporal dynamics on these static inputs. The model employs a combination of sequence-to-sequence layers for local processing and an interpretable multi-head attention mechanism to capture long-term dependencies, offering a comprehensive understanding of both short and long-term temporal relationships. By generating prediction intervals, TFT provides valuable insights into the possible range of future values, enhancing decision-making processes with a clearer assessment of risk and uncertainty.

Fabian et.al. [160] examines the importance of accurate thermal load forecasting for district heating and cooling networks and evaluates the performance of the Temporal Fusion Transformer (TFT) in this context, presenting its use for producing 72-hour heating load forecasts for three different district heating grids in the city of ULM. Comparing TFT's performance with other machine learning methods, superior forecasting abilities across various scenarios, significantly in the spring and fall seasons, was demonstrated. This improvement is attributed to TFT's attention-based mechanism, which excels in handling the temporal nature of the data and its ability to generalize across different conditions. The research underscores TFT's potential in optimizing the use of renewable energy and reducing reliance on fossil fuels in district heating systems. TFT consistently outperformed other methods in terms of Mean Absolute Percentage Error (MAPE) across all district heating networks. The study found that, in the spring, TFT's MAPE improvement ranged from 2% better for one network to 8% better for another, highlighting its robustness even in harder-to-predict seasons.

Ratchakit et al. [159] applies TFT to forecast vital sign trajectories in intensive care patients, focusing on heart rate (HR), respiratory rate (RR), and oxygen saturation (SpO2). The results show that TFT could effectively forecast vital sign trajectories, such as heart rate (HR) and respiratory rate (RR), in intensive care patients. The model could provide accurate future vital signs predictions, with most unseen values falling within the 95% prediction interval. The study highlights TFT's ability to capture temporal dynamics and potential in detecting irregular patterns in vital sign time series, suggesting its usefulness in clinical settings for early detection of patient deterioration.

Behrens et al. [160] explored TFT for thermal load prediction in district heating and cooling networks, providing a comparison with other machine learning methods and demonstrating its effectiveness in forecasting heating load for different grids. Liao & Radhakrishnan [162] tested the TFT approach for short-term load forecasting in power distribution networks, showing its effectiveness over traditional methods.

4.4. InParformer

InParformer [150] is another model based on transformer architecture for long-term time series forecasting. The architecture, shown in Figure 8, It features an interactive parallel attention mechanism for learning dependencies in both frequency and time domains, enhanced by query selection, key-value pair compression, and evolutionary seasonal-trend decomposition modules. These innovations target the challenges of redundancy, semantic density, and complex temporal patterns in time series data. The methodology emphasizes efficiency and interpretability, significantly outperforming state-of-the-art models across various real-world datasets.

InParformer demonstrates remarkable performance in long-term time series forecasting (LTSF) across various datasets and metrics. This performance is highlighted by its comparison with other state-of-the-art models such as FEDformer, Autoformer, Informer, and others, offering a comprehensive view of its capabilities. InParformer consistently outperformed competing models across multiple datasets, including ETT (Electricity Transformer Temperature), Electricity, Exchange, and Weather datasets, showcasing its versatility and robustness in handling different types of time series data. The model achieved significant reductions in Mean Square Error (MSE) and Mean Absolute Error (MAE), indicating its precise forecasting ability. For instance, in the ETTm2 dataset, InParformer achieved an MSE of 0.260 and an MAE of 0.323 for a prediction length of 192, outperforming FEDformer, which had an MSE of 0.269 and an MAE of 0.328 for the same prediction length.

Similarly, in the Exchange dataset, InParformer outperformed other models with an MSE reduction of up to 15.1% compared to FEDformer, highlighting its efficiency in datasets lacking clear periodicity.

These results underscore InParformer's advanced design, incorporating interactive parallel attention and evolutionary seasonal-trend decomposition, which enables it to capture complex temporal dependencies more effectively than its counterparts. Its superior performance across diverse forecasting horizons further emphasizes its stability and adaptability in varying temporal resolutions.

4.5. Informer

The Informer model, shown in Figure 9, is designed to handle the high prediction capacity required for capturing long-range dependencies between input and output efficiently. Informer addresses several problems with the traditional transformer model, such as quadratic time complexity, high memory usage, and limitations of the encoder-decoder architecture. To overcome these, Informer introduces three key innovations: (i) a ProbSparse self-attention mechanism that reduces time complexity and memory usage to O(L log L) while maintaining performance, (ii) self-attention distilling that emphasizes dominant attention and manages extremely long input sequences effectively, and (iii) a generative style decoder that predicts long time series sequences in one forward operation, significantly speeding up inference for long-sequence predictions. The Informer model demonstrates superior performance over existing methods through extensive experiments on four large-scale datasets [156].

An important variant of the Informer architecture is the Frequency Enhanced Decomposed Transformer, FedFormer, which aims to improve long-term series forecasting by combining a transformer model with seasonal-trend decomposition with a frequency enhancement model to handle short-term details. FEDformer is shown to be more effective and efficient than the standard transformer, with a linear complexity to sequence length, and it reduces prediction error significantly on benchmark datasets [163]. However, the Informer's distinctive characteristics, particularly its ProbSparse self-attention mechanism and generative style decoder, are unique solutions to the specific challenges of modeling long-term dependencies, and this are not addressed by the FEDformer approach.

4.6. Sageformer

The Series-Aware Framework for Long-Term Multivariate Time Series Forecasting architecture, known as SageFormer, introduces a novel framework for forecasting multivariate time series (MTS) data. MTS data are quite common with the rise of Internet of Things (IoT) devices. These devices generate vast amounts of MTS data, necessitating advanced forecasting models capable of understanding the intricate interplays and temporal dynamics within this data. Long-term fore casting of MTS data is particularly challenging due to the need to capture both intra- and inter-series dependencies accurately.

SageFormer, shown in Figure 10, leverages graph structures to discern and model complex relationships between different series, capturing diverse temporal patterns while filtering out redundant information. The framework integrates seamlessly with existing transformer-based models, enhancing their ability to understand inter-series relationships. This integration enriches the models without significantly increasing complexity. Through extensive experiments on real-world and synthetic datasets, SageFormer demonstrates superior forecasting performance compared to contemporary state-of-the-art approaches.

Unlike in traditional Transformer architecture where input token are obtained by projecting input time series in a patch, the Sageformer integrates global tokens to enhance series awareness [152]. It uses an iterative message-passing process shown in Figure 11. Graph Structure Learning employs end-to-end learning of the adjacency matrix to capture relationships across series without prior knowledge, making it versatile for different datasets. Experiments on six real-world datasets (e.g., Traffic, Electricity, Weather) and two synthetic datasets, were conducted demonstrating SageFormer's effectiveness across various domains. SageFormer outperforms nine popular models for long-term MTS forecasting models, including models that focus on inter-series dependencies and long-term context using transformers.

4.7. Autoformer

The Autoformer approach [148] is a variation of the transformer architecture that includes a deep decomposition architecture, as shown in Figure 12. The Autoformer consists of an inner series decomposition block, an autocorrelation mechanism, and a corresponding encoder and decoder. The Autoformer features an autocorrelation mechanism inspired by stochastic process theory, which focuses on the periodicity of the series to discover dependencies and aggregate representations at the sub-series level. This mechanism is more efficient and accurate than traditional self-attention, particularly for long-term forecasting tasks.

The Autoformer achieved state-of-the-art accuracy, with a 38% relative improvement over existing methods on six benchmark datasets that span five practical applications, including energy, traffic, economics, weather, and disease [164]. These datasets included (i) load and oil temperature data from an electric transformer, (ii) an electricity dataset that contains the hourly electricity consumption, (iii) exchange records of the daily exchange rates of eight different countries, (iv) hourly traffic data from California Department of Transportation, (v) weather recorded every 10 minutes for the year 2020 containing 21 meteorological indicators, and (vi) weekly recorded influenza like illness (ILI) patients data from Centers for Disease Control and Prevention of the United States. For the multivariate setting, Autoformer achieved state of the art performance for all benchmarks and all prediction length settings. Autoformer gave a 74% MSE reduction in ETT, 18% in electricity, 61% in exchange, 15% in traffic and 21% in weather. For the input 36-predict-60 setting of ILI, Autoformer delivered a 43% MSE reduction. Overall, Autoformer yielded a 38% averaged MSE reduction.

4.8. Pyraformer

In Pyraformer, a novel pyramidal attention-based transformer is proposed to bridge the gap between capturing the long-range dependencies and achieving a low time and space complexity [154]. Specifically, a pyramidal attention mechanism is developed by passing messages based on attention in the pyramidal graph as shown in Figure 13.

The edges in this graph can be divided into two groups: the inter-scale and the intra-scale connections. The inter-scale connections build a multiresolution representation of the original sequence: nodes at the finest scale correspond to the time points in the original time series (e.g., hourly observations), while nodes in the coarser scales represent features with lower resolutions (e.g., daily, weekly, and monthly patterns). Such latent coarser scale nodes are initially introduced via a coarser-scale construction module. On the other hand, the intra-scale edges capture the temporal dependencies at each resolution by connecting neighboring nodes together. As a result, this model provides a compact representation for long-range temporal dependencies among far-apart positions by capturing such behavior at coarser resolutions, leading to a smaller length of the signal traversing path. Moreover, modeling temporal dependencies of different ranges at different scales with sparse neighboring intra-scale connections significantly reduces the computational cost.

At the heart of Pyraformer is its hierarchical attention mechanism shown in Figure 14, which processes data in a pyramidal fashion. This design reduces the computation required for long sequences by summarizing information at multiple scales and then integrating these summaries to capture long-range dependencies. By leveraging this pyramidal structure, Pyraformer significantly reduces the time and space complexity associated with processing long sequences. This efficiency makes it a practical choice for large-scale applications where computational resources are a limiting factor. The architecture's design is inherently adaptable, making it suitable for a wide range of applications beyond just text processing. It has shown promising results in time series forecasting, where capturing long-range dependencies is crucial for accurate predictions.

The Pyraformer model has been evaluated across multiple datasets to demonstrate its effectiveness and efficiency. It has been observed to deliver an improvement in MSE and MAE while tested with three different datasets: Electricity [165], ETTh1 and ETTm1 [151]. For ETTh1, Pyraformer decreased the

MSE by 24.8%, 28.9%, 26.2% respectively when the prediction length is 168, 336, and 720. In applications ranging from financial market prediction to weather forecasting and natural language tasks, Pyraformer's innovative approach offers a balance between performance and efficiency.

4.9. W-Transformers

The W-Transformer is a wavelet-based transformer framework that marks a significant advancement in univariate time series forecasting. This framework, shown in Figure 15, leverages the maximal overlap discrete wavelet transformation (MODWT) to decompose time series data, enabling the capture of non-stationary and long-range nonlinear dependencies. The W-Transformer framework is designed to tackle the challenges of forecasting non-stationary time series data, which is a common scenario in real-world applications. Non-stationarity in time series data, characterized by changes in mean and variance over time, poses significant challenges for traditional forecasting models.

W-Transformers address this challenge by incorporating wavelet transformations with the Transformer architecture, allowing for the efficient capture of both local and global temporal dependencies in the data. The MODWT is employed as a preprocessing step to decompose the time series data into various frequency components. This decomposition allows the W-Transformer to analyze the data at multiple resolutions, capturing the inherent multi-scale temporal dynamics. The wavelet transformation's ability to handle non-stationarity makes it an ideal choice for preprocessing time series data for forecasting tasks. The W-Transformer architecture exhibited superior performance in root mean square error (RMSE) on four different datasets as shown in Table 4.

4.10. FedFormer

Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting (FEDformer) [156], is a novel architecture for long-term series forecasting. Its architecture is shown in Figure 16. FEDformer combines transformer models with seasonal-trend decomposition and frequency domain analysis to enhance forecasting accuracy. By incorporating Fourier and wavelet transforms, FEDformer achieves linear computational complexity, outperforming state-of-the-art models in efficiency and accuracy across multiple datasets. The approach addresses the limitations of traditional transformer models in capturing global time series trends, offering significant improvements in multivariate and univariate forecasting tasks.

The FEDformer architecture introduces a dual-path design integrating both Fourier and wavelet transforms to enhance time series forecasting. This structure allows for efficient processing of long sequences by decomposing them into frequency components, enabling the model to capture both global and local temporal dependencies with reduced computational complexity. The innovative use of frequency-enhanced attention mechanisms in FEDformer facilitates a more effective and scalable approach to long-term forecasting tasks.

The FEDformer model's performance was evaluated using six datasets covering a range of real-world scenarios including energy, economics, traffic, weather, and disease. FEDformer outperformed all other models on the six benchmark datasets across all prediction horizons, with an overall 14.8% relative MSE reduction compared to Autoformer. Notably, for some datasets like Exchange and ILI, the improvement was even more significant, exceeding 20%. This showcases FEDformer's strength in long-term forecasting and its ability to handle data without clear periodicity effectively. In univariate time series forecasting, FEDformer achieved an overall 22.6% relative MSE reduction compared to Autoformer [153]. For certain datasets, such as traffic data, the improvement exceeded 30%. This further validates FEDformer's effectiveness in long-term forecasting. The model's dual-path structure, utilizing both Fourier and wavelet transforms (denoted as FEDformer-f and FEDformer-w), allows it to excel across different datasets by leveraging their complementary strengths.

4.11. CrossFormer++

CrossFormer++ [157], is an enhanced vision transformer leveraging cross-scale attention for improved performance in image classification, object detection, instance segmentation, and semantic segmentation tasks. It introduces a cross-scale embedding layer (CEL) and long-short distance attention (LSDA) for efficient feature processing across scales. Additionally, it addresses issues like self-attention map enlargement and amplitude explosion with a progressive group size (PGS) and an amplitude cooling layer (ACL), respectively. Extensive experiments demonstrate CrossFormer++'s superior performance across various tasks compared to existing models.

CrossFormer++ employs a pyramid structure that organizes the transformer model into four stages as shown in Figure 17. Each stage is designed to progressively refine the features extracted from the input image, allowing for a hierarchical representation that captures both local and global contextual information effectively.

At the beginning of each stage, a Cross-scale Embedding Layer (CEL) is utilized to generate input tokens. The CEL operates by sampling patches from the input image using four different kernel sizes, allowing it to capture features at multiple scales. This multi-scale approach enables the model to maintain a balance between computational efficiency and the ability to capture detailed feature information from various parts of the image.

Within each CrossFormer block, the Long Short Distance Attention (LSDA) module is a key component. LSDA is divided into Short Distance Attention (SDA) and Long Distance Attention (LDA) mechanisms. SDA focuses on building dependencies among neighboring embeddings, capturing local feature information efficiently. Conversely, LDA is responsible for establishing connections between embeddings that are far apart, enabling the model to integrate global contextual information. This dual attention mechanism allows CrossFormer++ to effectively process visual information across different spatial ranges.

To enhance the model's ability to understand the positional relationship between different tokens, CrossFormer++ incorporates a Dynamic Position Bias (DPB) module. This module adapts the relative position bias to accommodate variable image and group sizes, ensuring that positional information is accurately captured regardless of the input dimensions. This flexibility is crucial for tasks like object detection, where the input image size can vary significantly.

Two additional innovations in CrossFormer++ are the Progressive Group Size (PGS) and the Amplitude Cooling Layer (ACL). PGS addresses the varying attention needs at different layers of the model by adjusting the group size progressively. This ensures that local features are emphasized in early layers, while global features are prioritized in deeper layers. The ACL is introduced to manage the amplification of activation amplitudes across layers, which can destabilize training. By cooling down the amplitude, ACL helps maintain training stability and improve model performance.

On ImageNet data, CrossFormer++ models achieve a noticeable improvement in accuracy over existing vision transformers and their predecessors (CrossFormer models), with gains up to 0.8% in average accuracy across different model sizes [157]. For instance, CrossFormer++-B achieves 84.2% accuracy. CrossFormer++ significantly outperforms most existing vision transformers in object detection and instance segmentation tasks on the COCO 2017 dataset. CrossFormer++ surpasses CrossFormer by at least 0.5% average precision (AP). The semantic segmentation task on the ADE20K dataset exhibits greater performance gains over other architectures as the model size increases, indicating its effectiveness in dense prediction tasks.

4.12. Transformer-XL

Transformer-XL [161] introduces a novel approach to language modeling, enabling the capture of longer-term dependencies beyond fixed-length contexts. It achieves this through a segment-level recurrence mechanism and a new positional encoding scheme, significantly improving performance over traditional models like RNNs and vanilla transformers. Transformer-XL demonstrates its effectiveness across various datasets, significantly reducing perplexity and enhancing text generation quality. This model represents a substantial advancement in handling sequential data, offering promising applications in areas requiring nuanced understanding of long-term context.

Transformer-XL incorporates a recurrence mechanism at the segment level, allowing the model to carry over information from previous segments. This design enables the model to maintain a longer effective context without being limited by the fixed size of input segments. During training, hidden states from previous segments are reused as extended context for the current segment, enhancing the model's ability to capture long-term dependencies. This approach addresses both the limitations of fixed-length contexts and the context fragmentation problem, leading to improved modeling of longer sequences.

A crucial innovation in Transformer-XL is the introduction of relative positional encodings, which replace the absolute positional encodings used in standard transformers. This change is necessary to maintain the coherence of positional information when reusing hidden states across segments. Relative positional encodings allow the model to understand the relative positions of tokens within a sequence, enabling the reuse of states without causing confusion about the temporal order of events. This technique not only preserves temporal information but also allows for more flexible and efficient handling of sequence lengths.

Transformer-XL reduced the state-of-the-art (SoTA) perplexity from 20.5 to 18.3, on WikiText-103 [166], showcasing its superiority over previous models in capturing long-term dependencies in a large dataset with an average article length of 3.6K tokens [161]. On the enwik8 dataset [167] that contains 100M bytes of unprocessed Wikipedia text, Transformer-XL achieved new SoTA results, outperforming previous Transformer models and conventional RNN-based models by a significant margin. Notably, the 12-layer Transformer-XL matched the performance of a 64-layer network from a previous study with only 17% of the parameter budget, emphasizing its efficiency [161].

Similar to enwik8, text8 contains 100M processed Wikipedia characters created by lowering case the text and removing any character other than the 26 letters a through z, and space. Transformer-XL adapted the same model and hyper-parameters from enwik8, achieving the new SoTA from 1.13 to 1.08 [161]. Transformer-XL significantly improved the SoTA from 23.7 to 21.8 [161]on the One Billion Word dataset [168], indicating its generalizability and effectiveness in modeling both short and long sequences.

5. Conclusion

In this review, we have explored advanced machine learning models for modeling long-term context. We have focused on enhanced transformer architectures that can solve important challenges such as biomedical image segmentation, time series forecasting, and language modeling. Innovations such as the LogTrans architecture, which combine CNNs with transformers were shown to be superior to standard transformer architectures. Other architectures, like Autoformer and CrossFormer++, introduce mechanisms for capturing periodicity and cross-scale features. These advancements signify major strides in accuracy, efficiency, and applicability of transformers across different domains, showcasing their versatility and potential for future research and applications. Table 5 provides a comparison of key features and advancements of these architectures.

In this chapter we covered both classical and modern approaches to modeling long-term context. Characteristics like autocorrelation, trend, and seasonality in time series data across various domains were discussed. Classical methods such as autoregressive models, moving averages, and Box-Jenkins methodology, as well as modern techniques like RNNs, CNNs and LSTMs were discussed.

This review underscores the importance of capturing long-term dependencies in time series data. It highlights studies demonstrating the effectiveness of capturing these dependencies for accurate prediction and classification. Central to these models is the transformer architecture that allows the system to focus on relevant parts of the input sequence, effectively capturing long-term dependencies without the limitations of recurrent layers. This review also explores applications of transformer variations in signal processing for tasks such as forecasting, classification, and anomaly detection, showcasing architectures like LogTrans, Informer, and Temporal Fusion Transformers. It compares traditional methods and modern approaches, highlighting their applications, advantages, and disadvantages.

The future of attention-based models and transformer architectures are promising due to its emphasis on domain-specific adaptations, hybrid model development, and possible improvement in optimizations. We may expect advancements in transformer encoding techniques to capture temporal relationships more effectively. Authors have proposed such an approach of detecting rare events in extremely long time series data. Additionally, research will explore integrating established time series methods within transformer frameworks. Another focus will be on quantifying the uncertainty in forecasting problems, enabling more reliable decision support systems. Advancements in handling multivariate time series with transformers are another area that will unlock the analysis of complex interdependent systems. Research on optimizing computational efficiency will be equally important for deploying transformer based models in real-time as well as resource-constrained time series applications.

Acknowledgements

Portions of the material presented here is supported by the National Science Foundation under grant no. 2211841. The National Science Foundation is not responsible for the views expressed in this material. All opinions, findings, conclusions, and recommendations are those of the author(s).

References

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] P. S. Cowpertwait and A. V. Metcalfe, *Introductory time series with R*. Springer Science & Business Media, 2009.

[3] C. Chatfield and H. Xing, *The analysis of time series: an introduction with R*. CRC press, 2019.

[4] A. Struckov, S. Yufa, A. A. Visheratin, and D. Nasonov, “Evaluation of modern tools and techniques for storing time-series data,” *Procedia Comput. Sci.*, 2019, doi: 10.1016/j.procs.2019.08.125.

[5] S. Vishnu and M. Uma, “Financial Time Series Analysis and Forecasting with Statistical Inference and Machine Learning,” *Adv. Sci. Technol.*, vol. 124, pp. 418–425, 2023, doi: 10.4028/p-sp20ub.

[6] J. Wang and R. S. T. Lee, “Chaotic Recurrent Neural Networks for Financial Forecast,” *Am. J. Neural Netw. Appl.*, 2021, doi: 10.11648/J.AJNNA.20210701.12.

[7] M. Ghil *et al.*, “ADVANCED SPECTRAL METHODS FOR CLIMATIC TIME SERIES,” *Rev. Geophys.*, vol. 40, 2002, doi: 10.1029/2000RG000092.

[8] U. Helfenstein, “The use of transfer function models, intervention analysis and related time series methods in epidemiology.,” *Int. J. Epidemiol.*, vol. 20 3, pp. 808–15, 1991, doi: 10.1093/IJE/20.3.808.

[9] M. C. Kelley, “Acoustic absement in detail: Quantifying acoustic differences across time-series representations of speech data,” *ArXiv*, vol. abs/2304.06183, 2023, doi: 10.48550/arXiv.2304.06183.

[10] Y. Zhou and F. Zhang, "Packaging virtual image auxiliary generation algorithm based on large language model (LLM)," in *2023 8th international conference on communication and electronics systems (ICCES)*, 2023, pp. 1738–1743. doi: 10.1109/ICCES57224.2023.10192861.

[11] S. Talwar, "Dynamic Forecasting: Efficacy of Rolling Symmetric and Asymmetric GARCH Models," *South Asian J. Manag.*, vol. 23, p. 102, 2016.

[12] P. Bondon, "Prediction with incomplete past of a stationary process," *Stoch. Process. Their Appl.*, vol. 98, pp. 67–76, 2002, doi: 10.1016/S0304-4149(01)00116-8.

[13] S. Madadi, B. Mohammadi-ivatloo, and S. Tohidi, "Dynamic Line Rating Forecasting Based on Integrated Factorized Ornstein–Uhlenbeck Processes," *IEEE Trans. Power Deliv.*, vol. 35, pp. 851–860, 2020, doi: 10.1109/TPWRD.2019.2929694.

[14] G. C. Ray, "An algorithm to separate nonstationary part of a signal using mid-prediction filter," *IEEE Trans Signal Process*, vol. 42, pp. 2276–2279, 1994, doi: 10.1109/78.317850.

[15] E. C. Hall, G. Raskutti, and R. Willett, "Learning High-Dimensional Generalized Linear Autoregressive Models," *IEEE Trans. Inf. Theory*, vol. 65, pp. 2401–2422, 2019, doi: 10.1109/TIT.2018.2884673.

[16] R. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, pp. 987–1007, 1982, doi: 10.2307/1912773.

[17] A. M. Rather, "A prediction based approach for stock returns using autoregressive neural networks," *2011 World Congr. Inf. Commun. Technol.*, pp. 1271–1275, 2011, doi: 10.1109/WICT.2011.6141431.

[18] B. Loneck and I. Zurbenko, "Theoretical and Practical Limits of Kolmogorov-Zurbenko Periodograms with DiRienzo-Zurbenko Algorithm Smoothing in the Spectral Analysis of Time Series Data," *ArXiv Appl.*, 2020.

[19] Z. Sun, H. Lu, J. Chen, and J. Jiao, "An Efficient Noise Elimination Method for Non-stationary and Non-linear Signals by Averaging Decomposed Components," *Shock Vib.*, 2022, doi: 10.1155/2022/2068218.

[20] D. Lee, D. Lee, M. Choi, and J. Lee, "Prediction of Network Throughput using ARIMA," *2020 Int. Conf. Artif. Intell. Inf. Commun. ICAIIC*, pp. 1–5, 2020, doi: 10.1109/ICAIIC48513.2020.9065083.

[21] N. Garg, K. Soni, T. K. Saxena, and S. Maji, "Applications of AutoRegressive Integrated Moving Average (ARIMA) approach in time-series prediction of traffic noise pollution," *Noise Control Eng. J.*, vol. 63, pp. 182–194, 2015, doi: 10.3397/1/376317.

[22] M. Valipour, M. E. Banihabib, and S. M. Behbahani, "Parameters Estimate of Autoregressive Moving Average and Autoregressive Integrated Moving Average Models and Compare Their Ability for Inflow Forecasting," *J. Math. Stat.*, vol. 8, pp. 330–338, 2012, doi: 10.3844/JMSSP.2012.330.338.

[23] B. Sameh and M. Elshabrawy, "Seasonal Autoregressive Integrated Moving Average for Climate Change Time Series Forecasting," *Am. J. Bus. Oper. Res.*, 2023, doi: 10.54216/ajbor.080203.

[24] D. Pitfield, "Predicting Air-Transport Demand," *Environ. Plan. A*, vol. 25, pp. 459–466, 1993, doi: 10.1068/a250459.

[25] K. Dozie and C. C. Ibebuogu, "Decomposition with the Mixed Model in Time Series Analysis using Buys-Ballot Procedure," *Asian J. Adv. Res. Rep.*, 2023, doi: 10.9734/ajarr/2023/v17i2465.

[26] H. Hebbel and S. Heiler, "Trend and seasonal decomposition in discrete time," *Stat. Hefte*, vol. 28, pp. 133–158, 1987, doi: 10.1007/BF02932596.

[27] H. He, S. Gao, T. Jin, S. Sato, and X. Zhang, "A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction," *Appl Soft Comput*, vol. 108, p. 107488, 2021, doi: 10.1016/j.asoc.2021.107488.

[28] W. Sulandari, Suhartono, Subanar, and H. Utami, "Forecasting time series with trend and seasonal patterns based on SSA," *2017 3rd Int. Conf. Sci. Inf. Technol. ICSITech*, pp. 648–653, 2017, doi: 10.1109/ICSITECH.2017.8257193.

[29] R. Lacroix, "Short Term Analysis of Raw Data and Business Cycle Estimation - Part 1: Estimation and Tests (Analyse Conjoncturelle de Données Brutes et Estimation de Cycles Partie 1: Estimation et Tests) (French)," *Econom. Data Collect. Data Estim. Methodol. EJournal*, 2008, doi: 10.2139/SSRN.1679800.

[30] X. Zhang and R. Li, "A Novel Decomposition and Combination Technique for Forecasting Monthly Electricity Consumption," vol. 9, 2021, doi: 10.3389/fenrg.2021.792358.

[31] G. Kaiser, "Windowed Fourier Transforms," pp. 44–59, 2011, doi: 10.1007/978-0-8176-8111-1_2.

[32] S. Bradford, "Time-Frequency Analysis of Systems with Changing Dynamic Properties," 2006, doi: 10.7907/HMK7-FJ81.

[33] M. Kolawole, "Essential relational functions," pp. 3–36, 2002, doi: 10.1016/B978-075065773-0/50004-2.

[34] S. Vergura, M. Carpentieri, and V. Puliafito, "A time-frequency analysis of electrical users by means of Fourier and Wavelet transforms," *2016 IEEE 16th Int. Conf. Environ. Electr. Eng. EEEIC*, pp. 1–6, 2016, doi: 10.1109/EEEIC.2016.7555704.

[35] G. E. Box and G. M. Jenkins, "Some recent advances in forecasting and control," *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 17, no. 2, pp. 91–109, 1968.

[36] H. Haviluddin and R. Alfred, "Forecasting network activities using ARIMA method," *J. Adv. Comput. Netw.*, vol. 2, pp. 173–177, 2014, doi: 10.7763/JACN.2014.V2.106.

[37] samrad Jafarian-Namin, S. Ghomi, M. Shojaie, and S. Shavvalpour, "Annual forecasting of inflation rate in Iran: Autoregressive integrated moving average modeling approach," *Eng. Rep.*, vol. 3, 2021, doi: 10.1002/eng2.12344.

[38] J. C. Duarte, G. C. L. Cruz, G. O. Merli, and F. R. Echeverría, "A comparison of time series forecasting between artificial neural networks and box and jenkins methods," *Rev. Tec. Fac. Ing. Univ. Zulia*, vol. 27, pp. 146–160, 2004.

[39] M. Jamii, N. Oumidou, and M. Maaroufi, "Using the Box-Jenkins ARIMA Approach for Long-term Forecasting of CO2 Emissions in Morocco," *Proc. 2nd Int. Conf. Big Data Model. Mach. Learn.*, 2021, doi: 10.5220/0010737600003101.

[40] S. Gelper, R. Fried, and C. Croux, "Robust Forecasting with Exponential and Holt-Winters Smoothing," *Econom. Single Equ. Models EJournal*, 2007, doi: 10.2139/ssrn.1089403.

[41] J. W. Taylor and P. McSharry, “Short-Term Load Forecasting Methods: An Evaluation Based on European Data,” *IEEE Trans. Power Syst.*, vol. 22, pp. 2213–2219, 2007, doi: 10.1109/TPWRS.2007.907583.

[42] L. Luo-man, “Application of Exponential Smoothing in Time Series Analysis,” *J. Shenyang Norm. Univ.*, 2009.

[43] M. Neves and C. Cordeiro, “Exponential smoothing and resampling techniques in time series prediction,” *Discuss. Math. Probab. Stat.*, vol. 30, pp. 87–101, 2010, doi: 10.7151/DMPS.1122.

[44] A. J. Zavala and A. R. Messina, “Dynamic harmonic regression approach to wind power generation forecasting,” *2016 IEEE PES Transm. Distrib. Conf. Expo.-Lat. Am. PES TD-*, pp. 1–6, 2016, doi: 10.1109/TDC-LA.2016.7805684.

[45] A. Q. Miah, “Analysis of Financial Data,” pp. 303–324, 2016, doi: 10.1007/978-981-10-0401-8_16.

[46] K. Jha, N. Sinha, S. Arkatkar, and A. Sarkar, “MODELING GROWTH TREND AND FORECASTING TECHNIQUES FOR VEHICULAR POPULATION IN INDIA,” *Int. J. Traffic Transp. Eng.*, vol. 3, pp. 139–158, 2013, doi: 10.7708/IJTTE.2013.3(2).04.

[47] N. Wonu and R. U. Orlu, “Applicability of Time Series Analysis for Forecasting Senior Secondary Student Mathematics Achievement in Nigeria,” *Am. J. Math. Stat.*, vol. 9, pp. 203–214, 2019.

[48] S. Idrees, M. A. Alam, and P. Agarwal, “A Prediction Approach for Stock Market Volatility Based on Time Series Data,” *IEEE Access*, vol. 7, pp. 17287–17298, 2019, doi: 10.1109/ACCESS.2019.2895252.

[49] J. P. Rivera, “The Role of Stationarity in Business and Economic Research 1,” *J. Econ. Econ. Educ. Res.*, vol. 16, p. 173, 2015.

[50] C. Hu, “The Topological Properties of COVID-19 Global Activity Time Series Forecasting,” *2020 5th Int. Conf. Inf. Sci. Comput. Technol. Transp. ISCTT*, pp. 228–237, 2020, doi: 10.1109/ISCTT51595.2020.00047.

[51] R. Dean and W. Dunsmuir, “Dangers and uses of cross-correlation in analyzing time series in perception, performance, movement, and neuroscience: The importance of constructing transfer function autoregressive models,” *Behav. Res. Methods*, vol. 48, pp. 783–802, 2016, doi: 10.3758/s13428-015-0611-2.

[52] J. Olden and B. Neff, “Cross-correlation bias in lag analysis of aquatic time series,” *Mar. Biol.*, vol. 138, pp. 1063–1070, 2001, doi: 10.1007/S002270000517.

[53] D. S. Taylor, “Time-Series Analysis,” *West. J. Nurs. Res.*, vol. 12, pp. 254–261, 1990, doi: 10.1177/019394599001200210.

[54] P. Zhang, Y. Huang, S. Shekhar, and V. Kumar, “Exploiting Spatial Autocorrelation to Efficiently Process Correlation-Based Similarity Queries,” pp. 449–468, 2003, doi: 10.1007/978-3-540-45072-6_26.

[55] K. Stattegger, “Application of time series analysis to the tectonic analysis of disturbed rock sequences recorded from drill hole logs with examples from the paleozoic of Graz (Austria),” *J. Int. Assoc. Math. Geol.*, vol. 15, pp. 673–685, 1983, doi: 10.1007/BF01033231.

[56] E. Ghaderpour, W. Liao, and M. Lamoureux, "Antileakage least-squares spectral analysis for seismic data regularization and random noise attenuation," *Geophysics*, vol. 83, 2018, doi: 10.1190/GEO2017-0284.1.

[57] S. Baisch and G. Bokelmann, "Spectral analysis with incomplete time series: an example from seismology," *Comput. Geosci.*, vol. 25, pp. 739–750, 1999, doi: 10.1016/S0098-3004(99)00026-6.

[58] R. Hegger, H. Kantz, and T. Schreiber, "Practical implementation of nonlinear time series methods: The TISEAN package.," *Chaos*, vol. 9 2, pp. 413–435, 1998, doi: 10.1063/1.166424.

[59] M. Small, "Time series embedding and reconstruction," pp. 1–46, 2005, doi: 10.1142/9789812567772_0001.

[60] E. Bradley and H. Kantz, "Nonlinear time-series analysis revisited.," *Chaos*, vol. 25 9, p. 097610, 2015, doi: 10.1063/1.4917289.

[61] H. Kantz, "Nonlinear time series analysis — Potentials and limitations," pp. 213–228, 1996, doi: 10.1007/BFB0105440.

[62] Y. Zou, R. Donner, N. Marwan, J. Donges, and J. Kurths, "Complex network approaches to nonlinear time series analysis," *Phys. Rep.*, 2019, doi: 10.1016/J.PHYSREP.2018.10.005.

[63] E. Pereda, R. Quiroga, and J. Bhattacharya, "Nonlinear multivariate analysis of neurophysiological signals," *Prog. Neurobiol.*, vol. 77, pp. 1–37, 2005, doi: 10.1016/j.pneurobio.2005.10.003.

[64] S. A. A. Karim, B. A. Karim, F. Andersson, M. Hasan, J. Sulaiman, and R. Razali, "Predicting Malaysia business cycle using wavelet analysis," *2011 IEEE Symp. Bus. Eng. Ind. Appl. ISBEIA*, pp. 379–383, 2011, doi: 10.1109/ISBEIA.2011.6088841.

[65] T. Bartosch and J. Wassermann, "Wavelet Coherence Analysis of Broadband Array Data Recorded at Stromboli Volcano, Italy," *Bull. Seismol. Soc. Am.*, vol. 94, pp. 44–52, 2004, doi: 10.1785/0120020134.

[66] N. Masuda and Y. Okabe, "Time series analysis with wavelet coefficients," *Jpn. J. Ind. Appl. Math.*, vol. 18, pp. 131–160, 2001, doi: 10.1007/BF03167358.

[67] S. Schiff, "Resolving time-series structure with a controlled wavelet transform," *Opt. Eng.*, vol. 31, pp. 2492–2495, 1992, doi: 10.1117/12.60040.

[68] C. Torrence and G. Compo, "A Practical Guide to Wavelet Analysis.," *Bull. Am. Meteorol. Soc.*, vol. 79, pp. 61–78, 1998, doi: 10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2.

[69] Y. Ansari *et al.*, "A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading," *IEEE Access*, vol. 10, pp. 127469–127501, 2022, doi: 10.1109/ACCESS.2022.3226629.

[70] A. M. Aboussalah, Z. Xu, and C.-G. Lee, "What is the Value of the Cross-Sectional Approach to Deep Reinforcement Learning?," *Electronic*, 2020, doi: 10.2139/ssrn.3748130.

[71] S. Consoli, L. Pezzoli, and E. Tosetti, "Neural forecasting of the Italian sovereign bond market with economic news," *J. R. Stat. Soc. Ser. A Stat. Soc.*, vol. 185, pp. S197–S224, 2022, doi: 10.1111/rssa.12813.

[72] K. Lei, B. Zhang, Y. Li, M. Yang, and Y. Shen, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Syst Appl*, vol. 140, 2020, doi: 10.1016/J.ESWA.2019.112872.

[73] Y. Li, W. Zheng, and Z. Zheng, "Deep Robust Reinforcement Learning for Practical Algorithmic Trading," *IEEE Access*, vol. 7, pp. 108014–108022, 2019, doi: 10.1109/ACCESS.2019.2932789.

[74] C.-T. Chen, A.-P. Chen, and S.-H. Huang, "Cloning Strategies from Trading Records using Agent-based Reinforcement Learning Algorithm," *2018 IEEE Int. Conf. Agents ICA*, pp. 34–37, 2018, doi: 10.1109/AGENTS.2018.8460078.

[75] M. Chandralekha and N. Shenbagavadivu, "Performance Analysis Of Various Machine Learning Techniques To Predict Cardiovascular Disease: An Emprical Study," *Appl. Math. Inf. Sci.*, vol. 12, pp. 217–226, 2018, doi: 10.18576/AMIS/120121.

[76] D. Senthil and G. Suseendran, "Efficient time series data classification using sliding window technique based improved association rule mining with enhanced support vector machine," *Int. J. Eng. Technol.*, 2018, doi: 10.14419/IJET.V7I2.33.13890.

[77] S. Ougiaroglou, K. Diamantaras, and G. Evangelidis, "Exploring the effect of data reduction on Neural Network and Support Vector Machine classification," *Neurocomputing*, vol. 280, pp. 101–110, 2017, doi: 10.1016/j.neucom.2017.08.076.

[78] X. Yang, G. Zhang, J. Lu, and J. Ma, "A Kernel Fuzzy c-Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems With Outliers or Noises," *IEEE Trans. Fuzzy Syst.*, vol. 19, pp. 105–115, 2011, doi: 10.1109/TFUZZ.2010.2087382.

[79] S. Sathyamoorthy and E. Sivasankar, "A Clustering-based Framework for Fast Training of Classifiers," *2020 Int. Conf. Innov. Trends Inf. Technol. ICITIIT*, pp. 1–6, 2020, doi: 10.1109/ICITIIT49094.2020.9071543.

[80] D. L. Aguilar, M. A. M. Perez, O. Loyola-González, K. Choo, and E. Bucheli-Susarrey, "Towards an interpretable autoencoder: A decision tree-based autoencoder and its application in anomaly detection," *IEEE Trans. Dependable Secure Comput.*, 2022, doi: 10.1109/tdsc.2022.3148331.

[81] J. Park, V. R. Surabhi, P. Krishnamurthy, S. Garg, R. Karri, and F. Khorrami, "Anomaly Detection in Embedded Systems Using Power and Memory Side Channels," *2020 IEEE Eur. Test Symp. ETS*, pp. 1–2, 2020, doi: 10.1109/ETS48528.2020.9131596.

[82] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," *Proc. Int. Jt. Conf. Neural Netw. 2003*, vol. 3, pp. 1741–1745 vol.3, 2003, doi: 10.1109/IJCNN.2003.1223670.

[83] A. L. Alfeo, M. Cimino, G. Manco, E. Ritacco, and G. Vaglini, "Using an autoencoder in the design of an anomaly detector for smart manufacturing," *Pattern Recognit Lett*, vol. 136, pp. 272–278, 2020, doi: 10.1016/j.patrec.2020.06.008.

[84] J. Yang, X. Yang, and Z. Zhang, "A High-dimensional Anomaly Detection Algorithm Based on IForest with Autoencoder," *2022 4th Int. Conf. Data-Driven Optim. Complex Syst. DOCS*, pp. 1–5, 2022, doi: 10.1109/DOCS55193.2022.9967746.

[85] N. Manaswi, "RNN and LSTM," pp. 115–126, 2018, doi: 10.1007/978-1-4842-3516-4_9.

[86] C.-H. Wu, C.-C. Lu, Y.-F. Ma, and R.-S. Lu, "A New Forecasting Framework for Bitcoin Price with LSTM," *2018 IEEE Int. Conf. Data Min. Workshop ICDMW*, pp. 168–175, 2018, doi: 10.1109/ICDMW.2018.00032.

[87] C. Luo and H. Wang, "Fuzzy forecasting for long-term time series based on time-variant fuzzy information granules," *Appl Soft Comput*, vol. 88, p. 106046, 2020, doi: 10.1016/j.asoc.2019.106046.

[88] H. J. Kim, A. C. Depoian, C. P. Bailey, and P. Guturu, "Novel neural network architecture for energy prediction," vol. 12097, pp. 1209705-1209705–5, 2022, doi: 10.1117/12.2619143.

[89] L. Chen and M. Xu, "Piecewise Time Series Prediction Based on Stacked Long Short-Term Memory and Genetic Algorithm," *2020 Chin. Autom. Congr. CAC*, pp. 519–525, 2020, doi: 10.1109/CAC51589.2020.9327694.

[90] U. Onyekpe, V. Palade, S. Kanarachos, and S. Christopoulos, "A Quaternion Gated Recurrent Unit Neural Network for Sensor Fusion," *Inf.*, vol. 12, p. 117, 2021, doi: 10.3390/info12030117.

[91] C. Tallec and Y. Ollivier, "Can recurrent neural networks warp time?," *ArXiv*, vol. abs/1804.11188, 2018.

[92] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions," *Procedia Comput. Sci.*, vol. 131, pp. 895–903, 2018, doi: 10.1016/J.PROCS.2018.04.298.

[93] W. Zheng and G. Chen, "An Accurate GRU-Based Power Time-Series Prediction Approach With Selective State Updating and Stochastic Optimization," *IEEE Trans. Cybern.*, vol. 52, pp. 13902–13914, 2021, doi: 10.1109/TCYB.2021.3121312.

[94] N. B. Erichson, S. H. Lim, and M. W. Mahoney, "Gated Recurrent Neural Networks with Weighted Time-Delay Feedback," *ArXiv*, vol. abs/2212.00228, 2022, doi: 10.48550/arXiv.2212.00228.

[95] R. Dangovski, L. Jing, P. Nakov, M. Tatalović, and M. Soljačić, "Rotational Unit of Memory: A Novel Representation Unit for RNNs with Scalable Applications," *Trans. Assoc. Comput. Linguist.*, vol. 7, pp. 121–138, 2019, doi: 10.1162/tacl_a_00258.

[96] M. Morchid, "Parsimonious memory unit for recurrent neural networks with application to natural language processing," *Neurocomputing*, vol. 314, pp. 48–64, 2018, doi: 10.1016/j.neucom.2018.05.081.

[97] M. Bilkhu, S. Wang, and T. Dobhal, "Attention is all you need for Videos: Self-attention based Video Summarization using Universal Transformers," *ArXiv*, vol. abs/1906.02792, 2019.

[98] B. Hong, Z. Yan, Y. Chen, and Xiaobo-Jin, "Long Memory Gated Recurrent Unit for Time Series Classification," *J. Phys. Conf. Ser.*, vol. 2278, 2022, doi: 10.1088/1742-6596/2278/1/012017.

[99] S. Som, N. Chandra, L. Ahuja, S. Khatri, and H. Monga, "Utilizing Gated Recurrent Units to Retain Long Term Dependencies with Recurrent Neural Network in Text Classification," vol. 2, pp. 89–102, 2021, doi: 10.52547/JIST.9.34.89.

[100] C.-L. Liu, W. Hsaio, and Y.-C. Tu, "Time Series Classification With Multivariate Convolutional Neural Network," *IEEE Trans. Ind. Electron.*, vol. 66, pp. 4788–4797, 2019, doi: 10.1109/TIE.2018.2864702.

[101] T. Nakamura, R. Shimizu, Y. Uwate, and Y. Nishio, "Time Series Analysis with Noise-Mixing Effects Using Neural Networks," *2022 19th Int. SoC Des. Conf. ISOCC*, pp. 269–270, 2022, doi: 10.1109/ISOCC56007.2022.10031334.

[102] R. Younis, S. Zerr, and Z. Ahmadi, "Multivariate Time Series Analysis: An Interpretable CNN-based Model," *2022 IEEE 9th Int. Conf. Data Sci. Adv. Anal. DSAA*, pp. 1–10, 2022, doi: 10.1109/DSAA54385.2022.10032335.

[103] G. S. Chadha, J. Kim, A. Schwung, and S. Ding, "Permutation Learning in Convolutional Neural Networks for Time-Series Analysis," pp. 220–231, 2020, doi: 10.1007/978-3-030-61609-0_18.

[104] N. I. Chervyakov, P. Lyakhov, M. Deryabin, N. Nagornov, M. Valueva, and G. Valuev, "Residue Number System-Based Solution for Reducing the Hardware Cost of a Convolutional Neural Network," *Neurocomputing*, vol. 407, pp. 439–453, 2020, doi: 10.1016/j.neucom.2020.04.018.

[105] A. B. P. Utama, A. Wibawa, M. Muladi, and A. Nafalski, "PSO based Hyperparameter tuning of CNN Multivariate Time- Series Analysis," *J. Online Inform.*, 2022, doi: 10.15575/join.v7i2.858.

[106] F. Jiang, X. Han, W. Zhang, and G. Chen, "Atmospheric PM2.5 Prediction Using DeepAR Optimized by Sparrow Search Algorithm with Opposition-Based and Fitness-Based Learning," *Atmosphere*, 2021, doi: 10.3390/ATMOS12070894.

[107] M. Dong *et al.*, "Deformation Prediction of Unstable Slopes Based on Real-Time Monitoring and DeepAR Model," *Sensors*, vol. 21, 2020, doi: 10.3390/s21010014.

[108] Y. Jeon and S. Seong, "Robust recurrent network model for intermittent time-series forecasting," *Int. J. Forecast.*, 2021, doi: 10.1016/j.ijforecast.2021.07.004.

[109] S. Park, S. Park, and E. Hwang, "Normalized Residue Analysis for Deep Learning Based Probabilistic Forecasting of Photovoltaic Generations," *2020 IEEE Int. Conf. Big Data Smart Comput. BigComp*, pp. 483–486, 2020, doi: 10.1109/BigComp48618.2020.00-20.

[110] L. Shen, Z. Wei, and Y. Wang, "Determining the Rolling Window Size of Deep Neural Network Based Models on Time Series Forecasting," *J. Phys. Conf. Ser.*, vol. 2078, 2021, doi: 10.1088/1742-6596/2078/1/012011.

[111] S. Li, H. Huang, and W. Lu, "A Neural Networks Based Method for Multivariate Time-Series Forecasting," *IEEE Access*, vol. 9, pp. 63915–63924, 2021, doi: 10.1109/ACCESS.2021.3075063.

[112] A. Gouttes, K. Rasul, M. Koren, J. Stephan, and T. Naghibi, "Probabilistic Time Series Forecasting with Implicit Quantile Networks," *ArXiv*, vol. abs/2107.03743, 2021.

[113] D. D. Chuwang and W. Chen, "Forecasting Daily and Weekly Passenger Demand for Urban Rail Transit Stations Based on a Time Series Model Approach," *Forecasting*, 2022, doi: 10.3390/forecast4040049.

[114] T. Toharudin, R. Pontoh, R. Caraka, S. Zahroh, Y. Lee, and R. Chen, “Employing long short-term memory and Facebook prophet model in air temperature forecasting,” *Commun. Stat. - Simul. Comput.*, vol. 52, pp. 279–290, 2021, doi: 10.1080/03610918.2020.1854302.

[115] C. Saiktishna, N. S. V. Sumanth, M. M. S. Rao, and T. J, “Historical Analysis and Time Series Forecasting of Stock Market using FB Prophet,” *2022 6th Int. Conf. Intell. Comput. Control Syst. ICICCS*, pp. 1846–1851, 2022, doi: 10.1109/ICICCS53718.2022.9788231.

[116] Q. Huang, “Forecasting Stock Prices Using Multi-Macroeconomic Regressors Based on the Facebook Prophet Model,” *BCP Bus. Manag.*, 2022, doi: 10.54691/bcpbm.v25i.1762.

[117] S. Mahmud, “Bangladesh COVID-19 Daily Cases Time Series Analysis using Facebook Prophet Model,” *PSN Dis. Illn. Top.*, 2020, doi: 10.2139/ssrn.3660368.

[118] O. Mphale, N. Raffing, S. Sheikh, and L. Balasubramanian, “Time Series Forecasting of COVID-19 Mortality in SADC region with Facebook Prophet Model,” *2022 Int. Conf. Smart Appl. Commun. Netw. SmartNets*, pp. 1–6, 2022, doi: 10.1109/SmartNets55823.2022.9994010.

[119] S. N, P. B, S. T. M, S. K. B, and L. G, “Historical analysis and forecasting of stock market using fbprophet,” *South Asian J. Eng. Technol.*, 2022, doi: 10.26524/sajet.2022.12.43.

[120] M. Lu, “Vector autoregression (var) — an approach to dynamic analysis of geographic processes,” *Geogr. Ann. Ser. B Hum. Geogr.*, vol. 83, pp. 67–78, 2001, doi: 10.1111/j.0435-3684.2001.00095.x.

[121] R. Myers, R. Piggott, and W. Tomek, “Estimating Sources of Fluctuations in the Australian Wool Market: An Application of VAR Methods,” *Aust. J. Agric. Resour. Econ.*, vol. 34, pp. 242–262, 1990, doi: 10.1111/J.1467-8489.1990.TB00498.X.

[122] P. Alvarez-De-Toledo, A. C. Márquez, F. Núñez, and C. Usabiaga, “Introducing VAR and SVAR predictions in system dynamics models,” *Int J Simul Process Model*, vol. 4, pp. 7–17, 2008, doi: 10.1504/IJSPM.2008.020609.

[123] M. W. McCracken, M. T. Owyang, and T. Sekhposyan, “Real-Time Forecasting with a Large, Mixed Frequency, Bayesian VAR,” *Econom. Model. Cap. Mark. - Forecast. EJournal*, 2015, doi: 10.2139/SSRN.2673962.

[124] L. Kilian and H. LÃ¼tkepohl, “Structural Vector Autoregressive Analysis,” 2017, doi: 10.1017/9781108164818.

[125] A. Galicia, R. L. Talavera-Llames, A. T. Lora, I. Koprinska, and F. Martínez-Álvarez, “Multi-step forecasting for big data time series based on ensemble learning,” *Knowl Based Syst*, vol. 163, pp. 830–841, 2019, doi: 10.1016/j.knosys.2018.10.009.

[126] P. Valatsos, T. Vafeiadis, A. Nizamis, D. Ioannidis, and D. Tzovaras, “Freight transportation route time prediction with ensemble learning techniques,” *25th Pan-Hell. Conf. Inform.*, 2021, doi: 10.1145/3503823.3503833.

[127] V. Derbentsev, V. Babenko, K. Khrustalev, H. Obruch, and S. Khrustalova, “Comparative Performance of Machine Learning Ensemble Algorithms for Forecasting Cryptocurrency Prices,” *Int. J. Eng.*, vol. 34, pp. 140–148, 2021, doi: 10.5829/IJE.2021.34.01A.16.

[128] C. Pop, V. Chifu, C. Cordea, E. S. Chifu, and O. Barsan, “Forecasting the Short-Term Energy Consumption Using Random Forests and Gradient Boosting,” *2021 20th RoEduNet Conf. Netw. Educ. Res. RoEduNet*, pp. 1–6, 2021, doi: 10.1109/RoEduNet54112.2021.9638276.

[129] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[130] J. J. Levy and A. J. O'Malley, "Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning," *BMC Med. Res. Methodol.*, vol. 20, 2019, doi: 10.1186/s12874-020-01046-3.

[131] J. Chen, "Models for Predicting Business Bankruptcies and Their Application to Banking and Financial Regulation," *CGN Corp. Gov. Bankruptcy*, 2019, doi: 10.2139/ssrn.3329147.

[132] F. Anifowose, C. Ayadiuno, and F. Reshedan, "Feature Selection Based Hybrid Machine Learning Approach to Formation Cementation Factor Prediction," *Day 3 Tue Oct. 15 2019*, 2019, doi: 10.2118/198074-ms.

[133] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke, "Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions," pp. 548–560, 2020, doi: 10.1007/978-3-030-44584-3_43.

[134] M. N. Sadat, X. Jiang, M. M. A. Aziz, S. Wang, and N. Mohammed, "Secure and Efficient Regression Analysis Using a Hybrid Cryptographic Framework: Development and Evaluation," *JMIR Med. Inform.*, vol. 6, 2018, doi: 10.2196/medinform.8286.

[135] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks," *Renew. Energy*, 2019, doi: 10.1016/J.RENENE.2018.10.031.

[136] E. Maiorana and P. Campisi, "Longitudinal Evaluation of EEG-Based Biometric Recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, pp. 1123–1138, 2017.

[137] T. Nakano, M. Miyasaka, T. Ohtaka, and K. Ohmori, "Longitudinal Changes in Computerized EEG and Mental Function of the Aged: A Nine-Year Follow-Up Study," *Int. Psychogeriatr.*, vol. 4, pp. 9–23, 1992.

[138] Y. Li *et al.*, "Hi-BEHRT: Hierarchical Transformer-Based Model for Accurate Prediction of Clinical Events Using Multimodal Longitudinal Electronic Health Records," *IEEE J. Biomed. Health Inform.*, vol. 27, pp. 1106–1117, 2021.

[139] J. Zhao, P. Papapetrou, L. Asker, and H. Boström, "Learning from heterogeneous temporal data in electronic health records," *J. Biomed. Inform.*, vol. 65, pp. 105–119, 2017.

[140] R. P. Thombs, X. Huang, and A. K. Jorgenson, "It's about time: How recent advances in time series analysis techniques can enhance energy and climate research," *Energy Res. Soc. Sci.*, vol. 72, p. 101882, 2021.

[141] J. A. Lutz, "The Evolution of Long-Term Data for Forestry: Large Temperate Research Plots in an Era of Global Change," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:85586126

[142] H. Kim, J.-T. Lee, K. C. Fong, and M. L. Bell, "Alternative adjustment for seasonality and long-term time-trend in time-series analysis for long-term environmental exposures and disease counts," *BMC Med. Res. Methodol.*, vol. 21, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:230657447

[143] S. T. Jackson, "Repurposing long-term ecological studies for climate change," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 120, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:263116326

[144] G. Chamberlain, "Longitudinal Analysis of Labor Market Data: Heterogeneity, omitted variable bias, and duration dependence," 1985. [Online]. Available: https://api.semanticscholar.org/CorpusID:151217097

[145] E. J. Lusk, "Time Series Forecasting in Stock Trading Markets," *Int. J. Res. Bus. Soc. Sci. 2147-4478*, 2019, [Online]. Available: https://api.semanticscholar.org/CorpusID:198663330

[146] D. R. Brillinger, "8 Analysis of variance and problems under time series models," *Handb. Stat.*, vol. 1, pp. 237–278, 1980.

[147] W. F. Velicer and P. C. M. Molenaar, "Time Series Analysis for Psychological Research," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:60610368

[148] A. Vaswani *et al.*, "Attention Is All You Need."

[149] "LogTrans: Providing Efficient Local-Global Fusion with Transformer and CNN Parallel Network for Biomedical Image Segmentation." Accessed: Sep. 04, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10074688/

[150] H. Cao, Z. Huang, T. Yao, J. Wang, H. He, and Y. Wang, "InParformer: Evolutionary Decomposition Transformers with Interactive Parallel Attention for Long-Term Time Series Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 6, Art. no. 6, Jun. 2023, doi: 10.1609/aaai.v37i6.25845.

[151] H. Zhou *et al.*, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," no. arXiv:2012.07436. arXiv, Mar. 28, 2021. doi: 10.48550/arXiv.2012.07436.

[152] Z. Zhang, X. Wang, and Y. Gu, "SageFormer: Series-Aware Graph-Enhanced Transformers for Multivariate Time Series Forecasting," no. arXiv:2307.01616. arXiv, Jul. 04, 2023. Accessed: Aug. 29, 2023. [Online]. Available: http://arxiv.org/abs/2307.01616

[153] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," no. arXiv:2106.13008. arXiv, Jan. 07, 2022. doi: 10.48550/arXiv.2106.13008.

[154] S. Liu *et al.*, "Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting," presented at the International Conference on Learning Representations, Oct. 2021. Accessed: Sep. 04, 2023. [Online]. Available: https://openreview.net/forum?id=0EXmFzUn5I

[155] L. Sasal, T. Chakraborty, and A. Hadid, "W-Transformers : A Wavelet-based Transformer Framework for Univariate Time Series Forecasting," no. arXiv:2209.03945. arXiv, Sep. 08, 2022. Accessed: Aug. 29, 2023. [Online]. Available: http://arxiv.org/abs/2209.03945

[156] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," no. arXiv:2201.12740. arXiv, Jun. 16, 2022. doi: 10.48550/arXiv.2201.12740.

[157] W. Wang *et al.*, "CrossFormer: A Versatile Vision Transformer Hinging on Cross-scale Attention," no. arXiv:2108.00154. arXiv, Oct. 08, 2021. doi: 10.48550/arXiv.2108.00154.

[158] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," *ArXiv*, vol. abs/1912.09363, 2019, doi: 10.1016/J.IJFORECAST.2021.03.012.

[159] R. Phetrittikun, K. Suvirat, T. N. Pattalung, C. Kongkamol, T. Ingviya, and S. Chaichulee, “Temporal Fusion Transformer for forecasting vital sign trajectories in intensive care patients,” *2021 13th Biomed. Eng. Int. Conf. BMEiCON*, pp. 1–5, 2021, doi: 10.1109/BMEiCON53485.2021.9745215.

[160] F. Behrens, S. Leiprecht, J. Brantl, and M. Finkenrath, “Temporal Fusion Transformer for thermal load prediction in district heating and cooling networks,” *Linköping Electron. Conf. Proc.*, 2022, doi: 10.3384/ecp192047.

[161] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *ArXiv Prepr. ArXiv190102860*, 2019.

[162] H. Liao and K. Radhakrishnan, “Short-Term Load Forecasting with Temporal Fusion Transformers for Power Distribution Networks,” *2022 IEEE Sustain. Power Energy Conf. ISPEC*, pp. 1–5, 2022, doi: 10.1109/iSPEC54162.2022.10033079.

[163] Z. Yang, L. Liu, N. Li, and J. Tian, “Time Series Forecasting of Motor Bearing Vibration Based on Informer,” *Sensors*, vol. 22, 2022, doi: 10.3390/s22155858.

[164] C. Yu, F. Wang, Z. Shao, T. Sun, L. Wu, and Y. Xu, “DSformer: A Double Sampling Transformer for Multivariate Time Series Long-term Prediction,” *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manag.*, 2023, doi: 10.1145/3583780.3614851.

[165] A. Trindade, “ElectricityLoadDiagrams20112014.” 2015.

[166] S. Merity, C. Xiong, J. Bradbury, and R. Socher, “Pointer sentinel mixture models,” *ArXiv Prepr. ArXiv160907843*, 2016.

[167] L. MultiMedia, “Large text compression benchmark,” 2009.

[168] C. Chelba *et al.*, “One billion word benchmark for measuring progress in statistical language modeling,” *ArXiv Prepr. ArXiv13123005*, 2013.

Table 1. Comparison of traditional methods

Model Name	Description	Application	Advantages	Disadvantages
Autoregressive (AR) Models	Models that use a linear combination of past values of the variable.	Economics, finance, weather forecasting.	Simple and effective for some types of time series data.	Assumes linearity and stationarity in data.
Moving Average (MA) Models	Models that use past forecast errors in a regression-like model.	Stock market analysis, sales forecasting.	Good for smoothing out noise and short-term fluctuations.	Limited to capturing only recent past influences.
ARMA Models	Combines AR and MA models to model time series data.	Signal processing, econometrics.	More flexible than pure AR or MA models.	Requires stationary data.
ARIMA Models	An extension of ARMA that includes differencing to make data stationary.	Financial market predictions, sales forecasting.	Effective for non-stationary data, including data with trends.	Model identification can be complex.
Seasonal Decomposition	Decomposes a time series into seasonal, trend, and residual components.	Seasonal data analysis in various fields.	Useful for understanding and modeling seasonal variations.	Assumes a repetitive seasonal pattern.
Fourier Analysis	Transforms time series into frequency components.	Signal processing, climatology.	Useful for identifying periodicities in data.	Not suitable for non-periodic or non-linear data.
Box-Jenkins Methodology	A systematic method of using ARIMA models.	Broad application in various time series analyses.	Provides a comprehensive approach to model building.	Requires expertise and can be time-consuming.
Exponential Smoothing	A method that weights the historical data, decreasing exponentially.	Inventory control, sales forecasting.	Simple to apply and effective for data with no clear trend or seasonality.	Struggles with data showing high variability or trends.
Trend Analysis	Identifying and analyzing trends in time series data.	Market analysis, environmental data analysis.	Useful for forecasting and understanding long-term trends.	Can oversimplify data by focusing mainly on trends.
Cross-Correlation and Autocorrelation Analysis	Measure the relationship between time series and their lags.	Signal processing, econometrics.	Useful for identifying lags of importance in time series data.	Limited in dealing with non-linear relationships.
Spectral Analysis	Analyzes the frequency spectrum in time series data.	Seismology, astronomy.	Effective in identifying dominant cycles and periodicities.	Requires understanding of advanced mathematical concepts.
Nonlinear Time Series Analysis	Methods to deal with nonlinear behaviors in time series.	Neuroscience, climate sciences.	Can capture complex dynamics not modeled by linear methods.	Often complex and require large amounts of data for modeling.
Wavelet Analysis	Breaking down data into different frequency components.	Signal processing, image analysis.	Good for analyzing data with time-varying frequencies.	Can be mathematically complex and computationally intensive.

Table 2. Comparison of modern approaches

Model Name	Description	Application Examples	Advantages	Disadvantages
Long Short-Term Memory (LSTM) Networks	RNNs capable of learning long-term dependencies in data.	Financial forecasting, speech recognition.	Good at capturing long-term dependencies in data.	Computationally intensive, prone to overfitting.
Gated Recurrent Units (GRUs)	Simplified version of LSTMs, also a type of RNN.	Natural language processing, music generation.	Require fewer parameters than LSTMs, faster training.	Less expressive than LSTMs for certain complex patterns.
Convolutional Neural Networks (CNNs) for Time Series	Utilize convolutional layers for time series data.	Image and signal processing, anomaly detection.	Effective in capturing spatial-temporal patterns.	Not inherently suited for sequence prediction tasks.
DeepAR	Probabilistic forecasting with autoregressive recurrent networks.	Demand forecasting, energy load forecasting.	Good for large datasets with multiple related series.	Requires large amounts of data to perform well.
Prophet	Designed for forecasting with daily observations.	Business metrics forecasting, web traffic.	Handles outliers, missing data, and seasonal effects.	Less effective for non-daily data or non-linear trends.
Vector Autoregression (VAR)	Captures linear interdependencies among multiple time series.	Econometrics, multivariate time series analysis.	Can model interdependencies in multiple time series.	Assumes linearity, not suitable for non-stationary data.
Ensemble Methods	Combines predictions from multiple models.	Financial time series prediction, weather forecasting.	Improves accuracy and robustness.	Can be complex to implement and interpret.
Hybrid Models	Combines traditional statistical models with machine learning.	Any application requiring both linear and non-linear modeling.	Captures both linear and non-linear aspects of data.	Can be complex to implement and tune.

Table 3. Ablation study of the effects of various component of LogTrans framework

Methods	Jaccard	Sensitiv-ity	mIoU	F1-Score
Backbone (EfficientNet-B6 + Concat + Decoder)	0.7744	0.8135	0.8422	0.8556
EfficientNet-B6 w/ Swin Transformer + Concat + Decoder	0.7746	0.815	0.8431	0.8552
EfficientNet-B6 w/ Swin Transformer + SeCo module + Decoder	0.7852	0.8257	0.8498	0.8638
EfficientNet-B6 w/ Swin Transformer + SeCo module + ReSD block + Decoder	0.7880	0.8343	0.8512	0.8661
Backbone (EfficientNet-B6 + Concat + Decoder)	0.7386	0.8352	0.8654	0.8297
EfficientNet-B6 w/ Swin Transformer + Concat + Decoder	0.7454	0.8394	0.8690	0.8346
EfficientNet-B6 w/ Swin Transformer + SeCo module + Decoder	0.7524	0.8582	0.8726	0.8421
EfficientNet-B6 w/ Swin Transformer + SeCo module + ReSD block + Decoder	0.7549	0.8450	0.8739	0.8442

Table 4. Comparison of W-Transformer with other architectures

Data	Metrics	WARIMA	ETS	SETAR	ARNN	RNN	Deep-AR	Transformer	W-Trans.
Website	RMSE	1281.64	1192.66	1082.51	1356.29	2593.36	2010.79	2638.05	847.41
Traffic	MAE	975.38	864.14	921.82	1065.48	2413.45	1875.34	2470.93	634.74
	sMAPE	39.48	36.31	43.89	41.23	164.07	107.14	180.14	31.02
	MASE	1.10	0.98	1.04	1.21	2.66	2.07	2.73	0.70
Sunspot	RMSE	41.48	37.46	57.06	71.83	74.16	52.50	40.63	30.07
	MAE	33.05	30.72	45.67	56.93	63.75	41.78	32.36	22.63
	sMAPE	41.48	38.21	62.91	97.60	108.69	65.21	40.40	30.09
	MASE	2.80	2.60	3.87	4.82	10.91	7.15	5.54	3.87
Japan	RMSE	196.65	186.15	297.30	239.31	171.51	179.61	326.55	76.21
Flu	MAE	174.17	171.63	281.93	199.93	114.01	163.67	276.56	58.98
	sMAPE	136.76	134.94	142.31	126.77	130.00	133.18	131.81	103.19
	MASE	4.83	3.95	6.49	4.60	2.27	3.26	5.51	1.17
Bangkok	RMSE	1889.92	3454.05	2153.80	819.90	824.70	786.21	767.52	735.00
Dengue	MAE	1756.66	3423.33	1486.24	678.36	681.73	634.59	611.18	608.30
	sMAPE	119.20	145.50	114.83	76.91	187.26	151.00	136.43	154.62
	MASE	7.57	14.75	6.40	2.92	2.56	2.38	2.29	2.28
Network	RMSE	43.94	23.65	40.58	24.71	43.00	22.51	29.21	19.00
Analytics	MAE	39.06	18.31	35.97	21.99	37.98	19.09	25.80	15.96
	sMAPE	94.56	70.46	91.69	75.80	93.34	71.52	80.64	60.31
	MASE	6.49	3.04	5.97	3.66	6.46	3.25	4.39	2.71

Table 5. Summary of transformer architectures

Model	Key Features	Application Areas	Notable Advancements
LogTrans	Dual-branch design with SeCo module	Biomedical image segmentation	Enhanced accuracy and robustness
TFT	Gated Residual Networks, LSTM, Multi-Head Attention	Time series forecasting	Superior forecasting abilities, handles missing data
InParformer	Interactive Parallel Attention	Long-term time series forecasting	Efficiency and interpretability in forecasting
Informer	ProbSparse self-attention, distilling	Long-term series forecasting	Reduced computational complexity, high performance
SageFormer	Graph structures for inter-series relationships	Multivariate time series forecasting	Enhanced forecasting performance
Autoformer	Decomposition architecture, Auto-correlation	Time series forecasting	Improved accuracy on periodicity and dependencies
Pyraformer	Pyramidal attention mechanism	Time series forecasting	Efficient long-range dependency capturing
W-Transformers	Wavelet-based preprocessing	Non-stationary time series forecasting	Effective capture of local and global dependencies
FEDformer	Seasonal-trend decomposition, frequency domain analysis	Long-term series forecasting	High efficiency and accuracy
CrossFormer++	Cross-scale attention mechanisms	Image classification and segmentation	Efficient processing of features across scales
Transformer-XL	Segment-level recurrence, relative positional encoding	Language modeling	Capture of longer-term dependencies, improved performance

Figure 1. Performance of Dow Jones from Jan 2023 to Feb 2024
(Source https://www.moneycontrol.com/us-markets/)

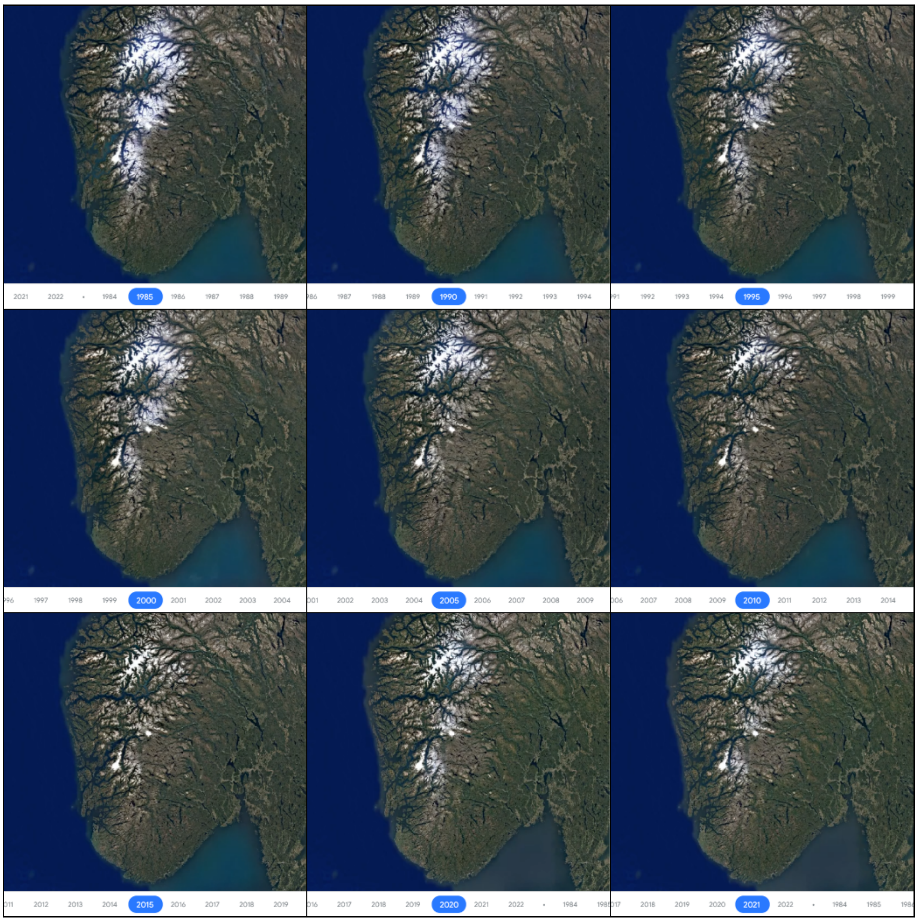

Figure 2. Norway's Ålfotbreen glacier is shrinking rapidly images from 1985(top left) to 2021(bottom right)

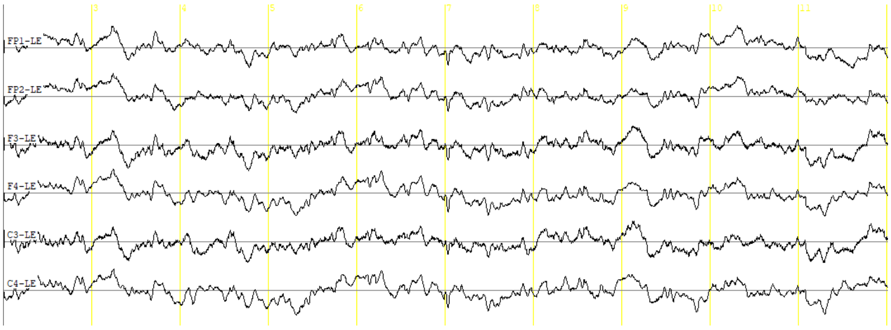

Figure 3. Recording of a 10 second EEG signal

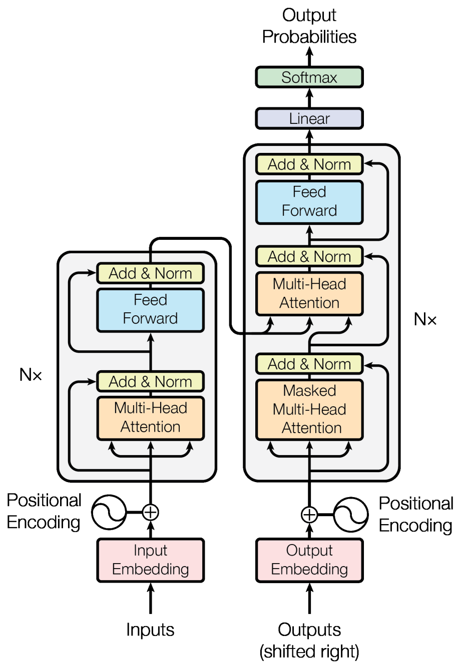

Figure 4. The original transformer model proposed in [148]

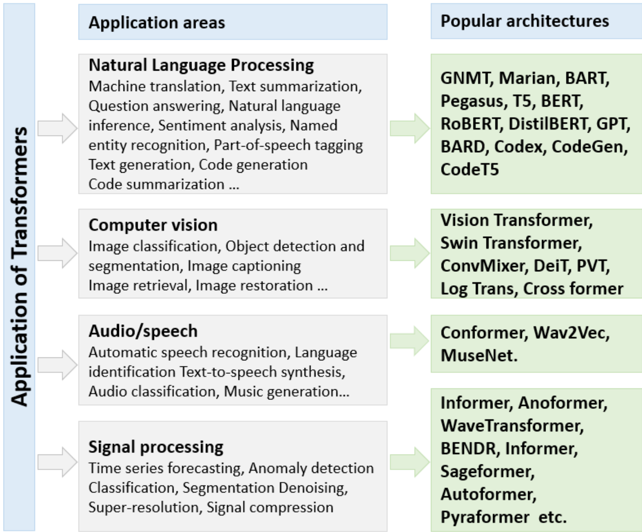

Figure 5. Popular transformer architectures and application areas

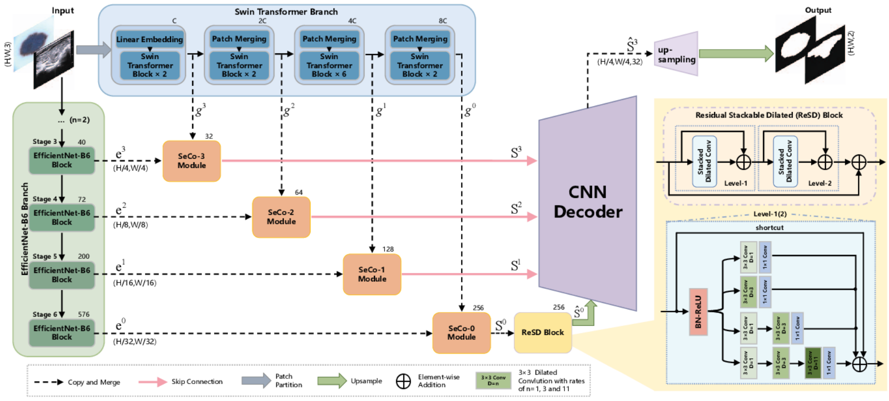

Figure 6. LogTrans Architecture [149]

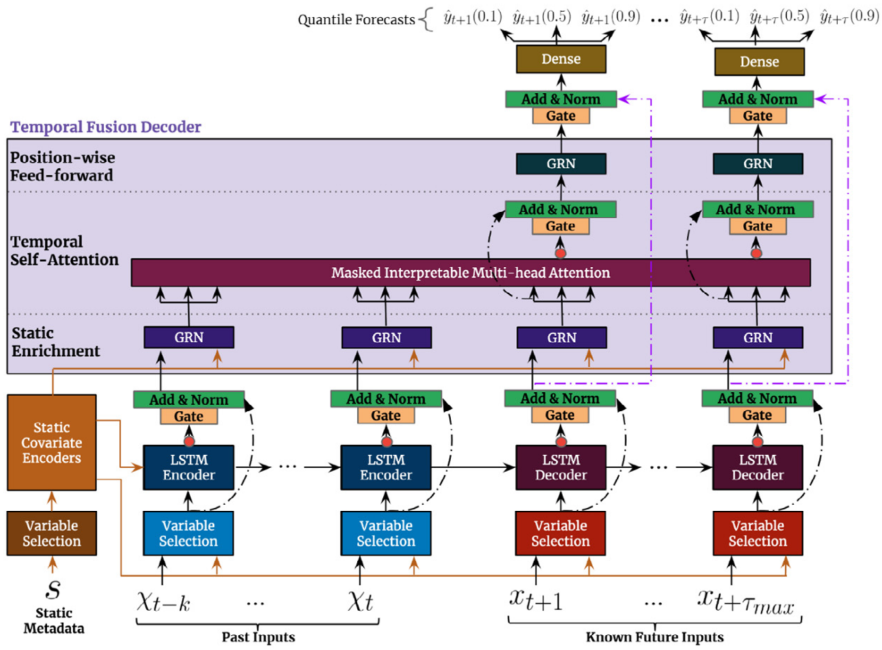

Figure 7. Temporal Fusion Transformer architecture

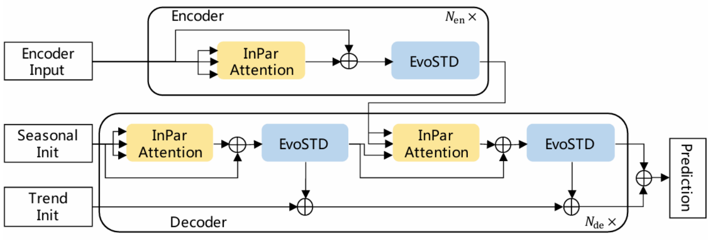

Figure 8. InParformer Architecture

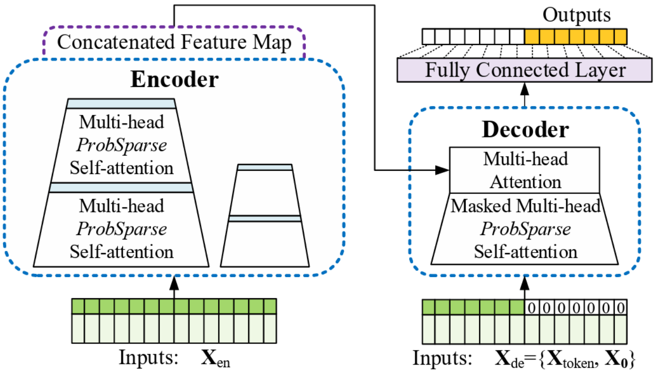

Figure 9. Informer model overview

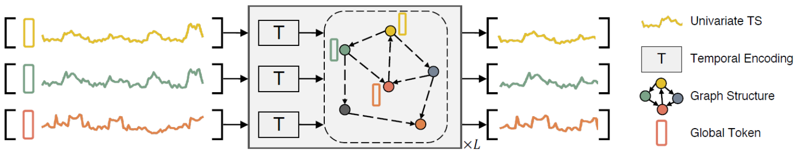

Figure 10. Series Aware Framework

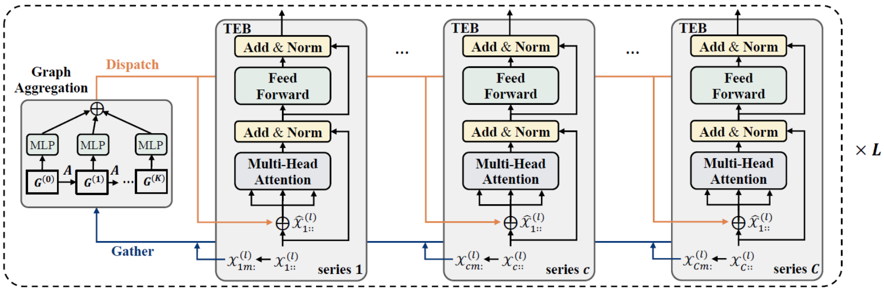

Figure 11. Illustration of the iterative message-passing process in SageFormer

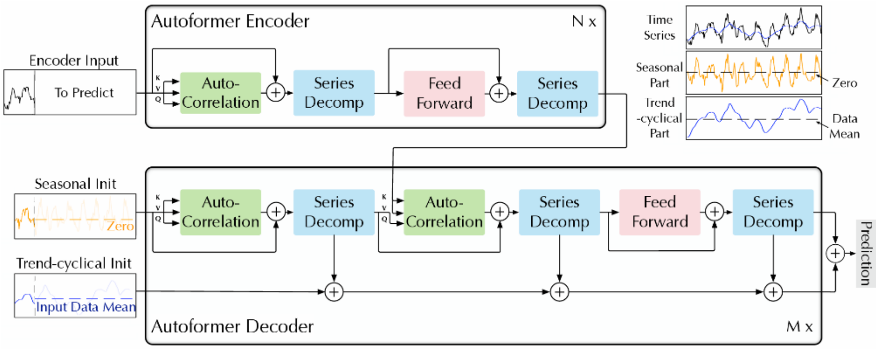

Figure 12. Autoformer architecture

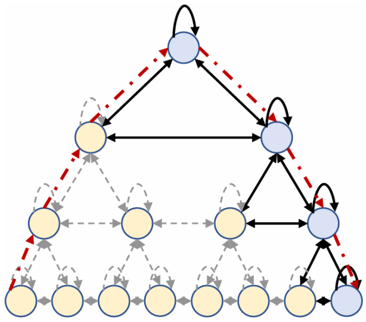

Figure 13. Pyramidal graph

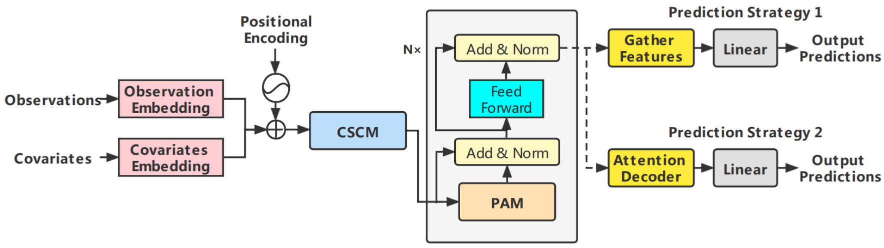

Figure 14. The architecture of Pyraformer

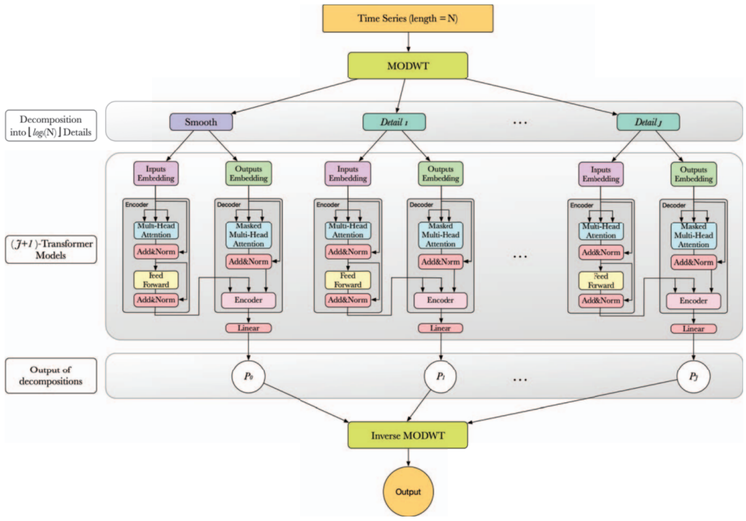

Figure 15. Architecture of W-Transformer

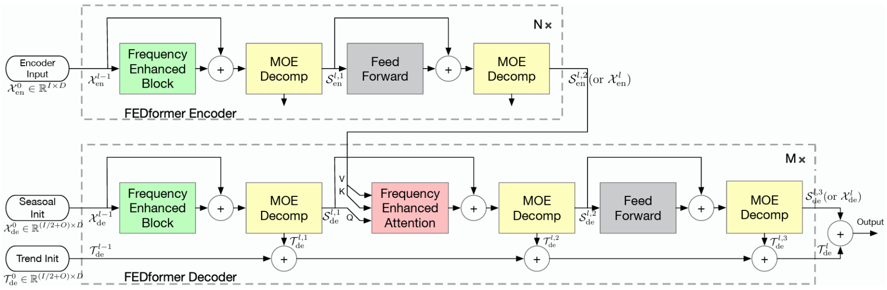

Figure 16. Fedformer Structure

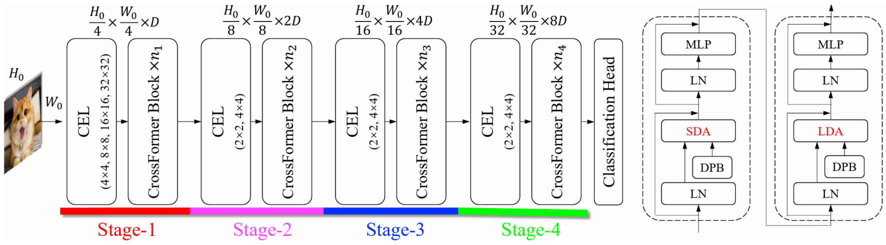

Figure 17. Architecture of Crossformer