Tushar Parulekar, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Comprehensive Survey of Advanced Time Series Signal Processing

Tushar Parulekar [1], Sandeep Chilukuri [2]
[1] Robert Bosch, USA Farmington Hills, USA
[2] Michigan Technological University Houghton, USA

Abstract- Time series data is observed in daily activities ranging from financial markets and automotive sensors to the medical industry and weather prediction. Proper analysis of this data plays a pivotal role in the era of artificial intelligence; with correct interpretation, we can utilize the data to its full potential. This article provides a holistic survey of the state of the art in time series signal processing, spanning from classical spectral decomposition and statistical filtering to the application of foundation models. We evaluate various architectures, including Convolutional Neural Networks (CNNs), Transformers, Graph Neural Networks (GNNs), and the emerging class of Structured State Space Models (SSMs) such as Mamba, specifically regard- ing their application to time series data. Additionally, we provide an overview of signal processing within deep learning contexts, exploring hybrid frameworks comprising wavelet transforms, Fourier analysis, and Kalman filtering. Finally, we assess the challenges faced in applying these concepts to time series data and discuss obstacles regarding the deployment of lightweight models.

Keywords—Time Series Analysis, Deep Learning, Signal Processing, Foundation Models.

I. INTRODUCTION

Time series data is ubiquitous in daily life, characterized by inherent dynamics and changes over time. Examples include weather patterns, sunlight intensity, and heart rate, all of which fluctuate dynamically throughout the day. Consequently, any data sequence recorded with respect to time is defined as time series data. Accurately interpreting this data is crucial for maximizing its utility in various applications that benefit human society.

For a long time, various mathematical techniques, such as Fourier transforms and the ARIMA framework, have been used to decompose time series to understand patterns, detect anomalies, and predict future events. While these methods function well with stable patterns, the continuous advancement in data acquisition techniques (sensors) has resulted in time series data that is highly dynamic (high frequency) and noisy. Consequently, the rigidity of traditional techniques often falls short in processing these signals, leading to information loss [1].

The application of Deep Learning (DL) has caused a paradigm shift in the field of time series analysis. By offering the opportunity to automatically extract features and learn non-linear behaviors, DL has significantly impacted the utilization of time series data in forecasting, classification, and anomaly detection. Despite these benefits, there is no consensus on an ”optimal” architecture suitable for all processing needs. Initially, DL employed Recurrent Neural Networks (RNNs), inspired by biological neural signal processing and designed for sequence retention. Subsequently, a more advanced method of interpreting time series data was developed: Temporal Convolutional Networks (TCNs). TCNs enabled the analysis of long data sequences simultaneously [2]. Following the robust reliance on neural networks, exploration expanded to Transformer architectures [3]. Most recently, the dominance of the Transformer has been challenged by efficient linear-complexity models like Mamba and simple MLP-based baselines, sparking a debate on the necessity of complex attention mechanisms for temporal data [2].

In this article, we hierarchically analyze the development of time series data analysis, beginning with classic signal processing techniques, moving through the evolution of neural networks, and finally discussing foundation models and Gen- erative AI. We also shed light on hybrid structures where the mathematical interpretability of signal processing meets the machine learning capabilities of neural networks.

II. TIME SERIES SIGNAL PROCESSING

Despite the prominence of modern deep learning techniques, many industries continue to rely on classical signal processing, and many machine learning architectures incorporate princi- ples of traditional signal processing into their neural logic.

A. Spectral Analysis and Frequency-Domain Processing

Spectral domain analysis has been a powerful paradigm for time series analysis, offering unique advantages over traditional time-domain approaches. It provides structured information about the time series signal that is often difficult to interpret in the time domain alone.

1) The Fourier Renaissance: Traditional Fourier Transforms (FT) assume that the signal is stationary within the processing duration (window). Consequently, they demonstrate limitations with evolving, dynamic time series signals, as they tend to lose information during conversion into the frequency domain. With the advancement of DL, architectures have begun using FT within their learning layers, elevating it from a mere preprocessing step to an embedding within the network. This allows DL networks to cater to both stationary and dynamic signals by learning optimal frequency representations during training, leading to a new field of Fourier-based methods in Artificial Intelligence (AI).

a) Fourier Basis Mapping (FBM): Recent research has identified limitations in standard Fourier-based DL methods, particularly their susceptibility to inconsistencies arising from variable starting cycles and differing series lengths. To address these challenges, the Fourier Basis Mapping (FBM) method was developed. FBM interprets the real and imaginary components of frequency spectra as coefficients for a set of cosine and sine basis functions, organized across tiered frequency levels. This approach facilitates the integration of time-frequency features through a Fourier basis expansion, mapping them into a unified, shared representational space. Consequently, FBM adeptly extracts explicit frequency characteristics while preserving crucial temporal properties. The efficacy of FBM has been demonstrated through its integration into various architectural backbones, including linear models, Multi-Layer Perceptrons (MLPs), and Transformer networks. This integration leverages "hard" frequency inductive biases, significantly stabilizing model training, especially for inherently periodic data [4].

b) Frequency-Aware MLPs: Beyond enhanced representational capabilities, the frequency domain offers substantial advantages in computational efficiency. Empirically, learning within the frequency domain can be computationally more efficient than its time-domain counterpart for a subset of tasks. Frequency-domain MLPs capitalize on the fundamental property that a convolution operation in the time domain is equivalent to element-wise multiplication in the frequency domain. These models project input data into the frequency domain via the Fast Fourier Transform (FFT). A learnable, complex-valued weighting is applied in this spectral domain before the data is transformed back into the time domain (via inverse FFT). This methodology enables global receptive fields across the input with a computational complexity of $O(N \log N)$, effectively circumventing the quadratic scaling inherent to Transformer architectures [5].

c) Wavelet Neural Networks: While FTs provide a global decomposition of time series into different frequencies, they operate at a fixed resolution, which can lead to improper interpretation of transient data and scaling issues. This limitation is addressed by Wavelet transforms, which use variable-sized windows to adapt to both time and frequency domains. This provides high frequency resolution for analyzing frequency components and temporal resolution for locating high-frequency components. This multi-scale analysis makes wavelet transforms a distinct option for non-stationary signals [5].

d) Multilevel Wavelet Decomposition Network (mWDN): This architecture integrates traditional signal processing into modern DL techniques. It leverages multi-level discrete wavelet decompositions to segment the time series signal into sub-series, each representing a distinct frequency range. mWDN processes the decomposed components individually but integrates them within the neural network. This facilitates the fine-tuning of global model parameters, such as weighting and filtering, via backpropagation. Thus, this technique retains wavelet transformation frequency information while autonomously identifying the most relevant frequency bands, reducing manual effort [6].

e) Wavelet-Encoded Image Time Series (WEITS): This architecture introduces double residuals integrated with wavelet decomposition to better capture multi-scale time series data. Here, the time series is decomposed into two main components: a long-term trend, representing the approximation coefficients, and a short-term trend, containing the high- frequency variations. This enables the DL network to learn the stable structure of the signal independent of noise. The outputs of the two processing paths are combined for the final prediction. This technique helps isolate long-term trends from short-term noise, enabling robust predictions [7].

B. Decomposition Methodologies

Decomposition is the process of separating a time series into its components, typically Trend, Seasonality, and Residual (Noise).

1) Empirical Mode Decomposition (EMD) and Variants: Unlike Fourier or Wavelet transforms, EMD is a data-driven method. It shifts the input signal to generate Intrinsic Mode Functions (IMFs). While powerful for non-linear data, EMD is prone to ”mode mixing” (where a single physical signal is split across multiple IMFs) and end effects [8].

a) Variational Mode Decomposition (VMD): This technique decomposes the input signal into different band- limited IMF’s and their respective central frequencies, thereby aiding in signal reconstruction. This method provides a more robust and theoretically sound decomposition. Consequently, it has seen significant application in hybrid deep learning models [30]. By separating high-frequency noise from the base signal, it provides a strong base signal-to-noise ratio for downstream neural networks.

b) The Data Leakage Challenge: A critical, often overlooked issue in decomposition-based forecasting is data leakage. Standard EMD or VMD algorithms operate on the entire sequence length. If applied naively during training, the decomposition at time t might be influenced by data at time t + k (future), leading to artificially inflated performance that

fails in production. The Moving Front (MF) method has been proposed to strictly enforce causality. It performs decompo- sition dynamically, ensuring that the components extracted at step t depend solely on history x0:t. Experiments on complex datasets like Indian Summer Monsoon Rainfall have shown that while MF is computationally more intensive, it is essential for valid evaluation [31].

C. Adaptive Filtering

State estimation involves inferring the hidden variables of a dynamical system from noisy observations.

1) Neural Kalman Filtering: The Kalman Filter (KF) stands as the optimal estimator for linear systems subjected to Gaussian noise. However, the inherent non-linearity and non-Gaussian characteristics prevalent in real-world phenomena often compromise its performance. The emerging field of Neural Kalman Filtering seeks to synthesize the KF’s interpretable ”Predict-Update” cycle with the powerful approximation capabilities of DL models.

a) Recursive KalmanNet: This architecture directly integrates a Kalman Filter within a Recurrent Neural Network (RNN) framework. A key innovation of Recursive KalmanNet is its ability to dynamically learn the process noise covariance (Q) and measurement noise covariance (R) matrices, parameters that are notoriously challenging to tune manually. It further enhances numerical stability by propagating error covariance using the Joseph formula and optimizing through Gaussian negative log-likelihood. This enables the model to effectively "learn how to filter" by adapting to specific noise characteristics, consistently outperforming conventional KFs in non-Gaussian noise environments [9].

b) Distributed and Hybrid Filters: For applications involving sensor networks, distributed Kalman approaches have been developed. In these systems, local neural networks approximate system dynamics at individual nodes, and a subsequent consensus mechanism efficiently fuses these local estimates. This distributed paradigm is particularly critical for edge computing scenarios where bandwidth limitations preclude the centralization of raw sensor data, thereby enabling localized processing and reduced communication overhead [10].

III. APPLICATION OF DEEP LEARNING ARCHITECTURES

Time series modeling has shifted towards a rich ecosys- tem of competing architectures, moving beyond traditional Recurrent Neural Networks (RNNs). This section analyzes the primary architectural families, their mechanisms, and their trade-offs.

A. Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs), including variants like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), are inherently designed to process time series data. Their architecture incorporates feedback loops, maintaining an internal "hidden state" that captures information from previous time steps. This mechanism enables RNNs to model temporal dependencies and learn patterns across sequences by iteratively processing data from $t \rightarrow t + 1$. However, this sequential processing often leads to challenges such as vanishing or exploding gradients for very long sequences and limits parallelization during training [32].

B. Temporal Convolutional Networks (TCNs)

While RNNs process data sequentially, their limitations are overcome using TCNs, which approach time series as a 1D spatial structure with time as the primary dimension. Their advantages are discussed below:

- Dilated Causal Convolutions: The central innovation within TCNs lies in the application of dilated causal convolutions. These convolutions utilize filters that skip input steps based on an exponentially increasing dilation factor (e.g., 1, 2, 4, 8). This mechanism allows the network's receptive field to expand exponentially with depth, effectively capturing extremely long historical dependencies without succumbing to the vanishing gradient problems common in RNNs. The "causal" aspect ensures that predictions at time t depend only on past observations.

- Parallelism: A significant advantage of TCNs over RNNs is their ability to process an entire input sequence in parallel during training. This parallelizable architecture translates into substantial speedups, particularly when leveraging modern GPU hardware.

- Performance: TCNs have consistently demonstrated competitive, and often superior, performance compared to LSTMs on tasks demanding long-term memory, thereby establishing them as a robust baseline for both time series forecasting and classification applications [2].

C. Transformer Architectures and Attention Mechanisms

Traditionally, Transformer architectures have seen strong application in language models, but they are increasingly utilized in time series data analysis. The

Transformer architecture, defined by the self-attention mechanism, has a strong application in learning dependencies in time series data channels. Theoretically, this allows for modeling dependencies between any two points in a time series, regardless of distance.

1) Adaptation of Transformer Architectures to Time Series: The application of Transformer architectures to time series data necessitates specific adaptations to the time domain, including:

- Patching (PatchTST): Instead of feeding individual time steps as tokens (which yields high redundancy and computational cost), modern Time Series Transformers segment the series into ”patches” (sub-sequences). This reduces the token count by a factor of the stride, enabling the processing of much longer histories. Patching also preserves local semantic patterns (e.g., the shape of a heartbeat) within the token embedding [11].

- Positional Encodings: Standard sinusoidal encodings are often insufficient. Learnable positional embeddings or relative positional biases are used to help the model distinguish between ”near” and ”far” history.

- Channel Independence vs. Mixing: A major debate in the field is how to handle multivariate data.

– Channel Mixing: Models like the original Trans- former or Crossformer project all variables into a single embedding space to learn cross-variate corre- lations.
– Channel Independence: Models like PatchTST and iTransformer treat each variable as an independent univariate series (sharing the same model weights). Empirical results suggest Channel Independence often outperforms mixing because it avoids overfitting to spurious correlations between variables, which are common in finite datasets [11].

2) iTransformer: The iTransformer (Inverted Transformer) proposes a radical structural shift. Instead of applying attention across time tokens, it applies attention across variate tokens. The entire time series for a single variable is embedded into one token. The attention mechanism then models the multivariate correlation, while a feed-forward network (MLP) handles the temporal evolution. This structure has achieved new state-of-the-art results for multivariate forecasting [13].

D. Graph Neural Networks (GNNs) for Spatiotemporal Data

In many domains, time series are not isolated but linked by an underlying structure (e.g., road networks, power grids).

- Spatiotemporal GNNs: These models combine graph convolution (for spatial dependencies) with TCNs or RNNs (for temporal dependencies). For example, in traf- fic forecasting, the graph captures how congestion flows from upstream to downstream nodes [33].
- Dynamic Graph Learning: A key limitation of early Graph Signal Processing (GSP) was the assumption of a static graph. Real-world relationships evolve. Dynamic GNNs learn a time-varying adjacency matrix, allowing the model to infer that the relationship between two sensors changes depending on the time of day or system state [15].
- Hierarchical GNNs (DeepHGNN): This framework ad- dresses hierarchical forecasting (e.g., predicting energy usage at the appliance, house, and neighborhood levels

simultaneously). It uses graph-based interpolation to pool information across hierarchy levels, ensuring that fore- casts are coherent (the sum of parts equals the whole) [14].

E. State Space Models (SSMs) and Mamba

The most significant recent development in time series processing is Structured State Space Models (SSMs), such as the Mamba architecture.

- The Linear Complexity Promise: SSMs scale linearly, compared to the quadratic scaling of Transformers. This allows SSMs to process sequences of length 10k or 100k, which is impossible for standard Transformers.

- Mamba Mechanism: Mamba utilizes a Selective Scan mechanism. Mamba allows its parameters to be input- dependent. This enables the model to selectively propa- gate or forget information based on the current context, solving the "forgetting" problem of RNNs while main- taining training parallelism.

• **Time Series Variants:**

– Time-MoE (Mixture of Experts): MoE uses the Mamba framework to allow scaling of the model input parameter count (up to 2.4 billion) without affecting inference speed, as a limited number of "experts" are active for any given query [16].

– Mamba-ProbTSF: This technique extends Mamba to probabilistic forecasting by using a dual-network structure. One Mamba network predicts the mean, while another estimates the variance, thus minimiz- ing divergence against the data distribution [18].

– TSMamba: This is a foundation model using forward and backward Mamba encoders. It employs a two- stage transfer learning process: optimizing the archi- tecture via patch-wise autoregression, followed by fine-tuning the predictions [17].

F. Architectural Comparison

Table I
Comparison Of Various Architectures.

Feature	RNN/ LSTM	TCN (CNN)	Transformer	Mamba (SSM)	GNN
Long-Range Dependency	Poor (Vanishing Grad.)	Good (Dilated)	Excellent	Excellent	N/A
Training Speed	Slow (Sequential)	Fast	Fast	Very Fast	Slow (If Dynamic)
Key Mechanism	Gating / Memory Cell	Dilated Convolution	Self-Attention	Selective State Space	Message Passing
Best Use Case	Short sequences, Online	Audio, Waveforms	Semantic/ Global context	Massive sequences	Networked systems

IV. FOUNDATION MODELS AND GENERALIST TIME SERIES INTELLIGENCE

With large language transformer models resolving major natural language processing issues, aggressive research is currently being conducted into the application of Transformer architectures for time series data. These applications are known as Time-Series Foundation Models (TSFMs). These models are pre-trained with large amounts of world data available from multiple sources.

A. Pre-training Strategies and Datasets

Building a TSFM requires a large amount of quality data to learn the overall structure of time series.

• **Datasets:** Some of the new large-scale archives include Time-300B (300 billion time points across 9 domains) and the Time Series Library (TSLib), which consolidates diverse benchmarks [16].

• **Pre-training Objectives:**

– Masked Modeling: Similar to BERT (used in GPT training), portions of the time series are masked, and the model is challenged to reconstruct them. This enables the model to learn time series structures and smoothness [9].

– Forecasting (Model Prediction): Similar to GPT, the model is expected to predict the next token in the sequence. Time-MoE and Chronos are used for this.

B. Time Series Foundation Models (TSFM)

There are two main types of time series foundation models currently in use:

- Time-LLM: This approach utilizes a frozen text-based LLM (like LLaMA) and trains on time series data. The time series data is mapped into the LLM's text embedding space. The hypothesis is to use the LLM's pre-trained pattern recognition and reasoning capabilities without training a completely new model. This technique has shown surprising efficacy, especially in zero-shot inference where the model leverages ”world knowledge” (e.g., understanding a spike in weather data) [21].

- Native Time Series FMs: In this approach, the time se- ries model is trained from scratch on only numerical data (e.g., MOIRAI, TimeGPT, Time-MoE). These models are optimized for continuous numerical distributions, which are the backbone of time series data [16].

C. Application of Time Series Foundation Models

TSFMs are expected to see widespread application in real-world scenarios as they enable not only the generation of new time series data but also the prediction and forecasting of events.

- Time Weaver: A diffusion-based model that generates time series. It can be used for data augmentation (gen- erating synthetic samples to balance a dataset) or data chunking. It uses time series generation for denoising pro- cesses, learning to construct new informative sequences from Gaussian noise conditioned on control signals [21].
- Probabilistic Generation: Models like Chronos can understand preceding time data and generate complete future data rather than a single trajectory, providing robust applications in forecasting events and anomaly detection [22].

V. EVALUATION METRICS AND CHALLENGES

With the advancement in time series processing techniques, it is unavoidable to develop rigorous evaluations to prevent overfitting, underfitting, and hallucinations.

A. Evaluation of Time Series

- Forecasting: Beyond MSE/MAE, which are traditionally used for basic neural networks, new metrics like SMAPE (Symmetric Mean Absolute Percentage Error) and MASE (Mean Absolute Scaled Error) are being used as standards to handle varying scales.

- Compression as Evaluation: A novel paradigm proposes using the Lossless Compression ratio as a proxy for modeling capacity. Based on Shannon’s source coding theorem, a better model of the data distribution should be able to compress the data more efficiently. The TSCom- Bench framework evaluates models based on their ability to minimize the negative log-likelihood (equivalent to compression length), revealing distributional weaknesses that point metrics miss [13].

- Denoising: As Signal-to-Noise Ratio (SNR) gains more attention, improved analysis in that domain is critical to finding the ground truth [37].

B. Challenges in Time Series Signal Processing

Despite fine-grained processing developments and capabilities, challenges remain regarding time series signals. Some common issues are discussed below.

1) Non-stationarity: All time series signals are non- stationary. A model trained on 2020 data may become invalid in 2021 due to changes in patterns or shifts in standardization. Current approaches often simply normalize the input data and use it for training. While this helps convergence, it destroys the ability to model the magnitude of the signal and its reconstruction capabilities. Future research must focus on models that can adapt their internal parameters online as the distribution shifts, rather than just normalizing the input [33].

2) Edge Deployment and Efficiency: While Foundation Models grow larger (2.4B parameters), real-world applications (IoT, Wearables) demand smaller models with focused use cases. Architectures like FITS (Frequency Interpolation Time Series) and TimeseriesMixer provide lightweight alternatives. FITS treats analysis as interpolation in the complex frequency domain, achieving state-of-the-art (SOTA) performance with a fraction of the parameter count

of Transformers. The inte- gration of these lightweight models with standard hardware represents the next frontier for time series [29].

3) Interpretability: Often, the output of time series process- ing is a black-box model; however, in fields like medicine and automotive, these black boxes are unacceptable. Consequently, models like N-BEATS and FEDformer produce outputs as a

sum of interpretable components (Trend + Seasonality). This structural constraint ensures that the user can inspect why a forecast was made. Using generative models to ask "What if?" questions will need to be addressed with strong prediction, as it is a growing area of research requiring causal integration [17].

VI. CONCLUSION

Time series signal processing is currently undergoing a major overhaul. The era of manual feature engineering has ended abruptly. The application of Deep Learning, integrating lessons from signal processing such as spectral analysis, decomposition, and state estimation, is actively being applied across various fields.

Furthermore, there is a distinct divergence in the field of foundation models. On one hand, there is a move towards massive, universal Foundation Models that leverage the scaling of large amounts of data into Transformer and Mamba architectures. On the other hand, there is an extensive need for refined Lightweight and Hybrid Models that prioritize efficiency, interpretability, and mathematical robustness for specific applications.

The future of time series analysis lies in the application of Foundation Models that are not only large but also structured with physics-informed logic and signal processing capabilities, enabling adaptation to the non-stationary, evolving reality of real-world data.

REFERENCES

1. Z. Z. Darban et al., "Deep learning for time series anomaly detection: A survey," arXiv preprint arXiv:2211.05244, 2024.
2. J. Kim, H. Kim, H. Kim, D. Lee, and S. Yoon, "A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges," arXiv preprint arXiv:2411.05793, 2024.
3. S. R. K. Kottapalli et al., "Foundation models for time series: A survey," arXiv preprint arXiv:2504.04011, 2025.
4. R. Yang, L. Cao, X. You, K. Fang, J. Li, and J. Yang, "Fourier basis mapping: A time-frequency learning framework for time series forecasting," arXiv preprint arXiv:2507.09445, 2025.
5. Y. Wang et al., "Deep time series models: A comprehensive survey and benchmark," arXiv preprint arXiv:2407.13278, 2024.
6. B. Lim and S. Zohren, "Deep learning for time-series forecasting: A survey," arXiv preprint arXiv:1806.08946, 2018.
7. Z. Guo, Y. Sun, and T. Wu, "WEITS: A wavelet-enhanced residual framework for interpretable time series forecasting," arXiv preprint arXiv:2405.10877, 2024.
8. N. Safari, G. Price, and C. Y. Chung, "Analysis of empirical mode decomposition-based load and renewable time series forecasting," arXiv preprint arXiv:2011.11410, 2020.
9. H. Mortada, C. Falcon, Y. Kahil, M. Clavaud, and J.-P. Michel, "Recur- sive KalmanNet: Deep learning-augmented Kalman filtering for state estimation with consistent uncertainty quantification," arXiv preprint arXiv:2506.11639, 2025.
10. Y. Li, M. Zhao, H. Li, and G. Hao, "The Centralized Fusion Unscented Kalman Filter for Nonlinear System with correlated noise," in Proc. IEEE Int. Conf. Meas., Control, Electron. and Instrum. (IMCEC), 2019, doi: 10.1109/IMCEC46724.2019.8983873.
11. C. Xu, D. Wu, Y. Zhu, and Y. N. Wu, "Multivariate long-term time series forecasting with Fourier neural filter," arXiv preprint arXiv:2506.09174, 2025.

12. X. Qiu et al., "A comprehensive survey of deep learning for multivariate time series forecasting: A channel strategy perspective," arXiv preprint arXiv:2502.10721, 2025.
13. M. Wan et al., "Lossless compression: A new benchmark for time series model evaluation," arXiv preprint arXiv:2509.21002, 2025.
14. A. Sriramulu, N. Fourrier, and C. Bergmeir, "DeepHGNN: Study of graph neural network based forecasting methods for hierarchically related multivariate time series," arXiv preprint arXiv:2405.18693, 2024.
15. Y. Zheng, L. Yi, and Z. Wei, "A survey of dynamic graph neural networks," arXiv preprint arXiv:2404.18211, 2024.
16. X. Shi et al., "Time-MoE: Billion-scale time series foundation models with mixture of experts," arXiv preprint arXiv:2409.16040, 2024.
17. H. Ma et al., "A Mamba foundation model for time series forecasting," arXiv preprint arXiv:2411.02941, 2024.
18. P. Pessoa, P. Campitelli, D. P. Shepherd, S. B. Ozkan, and S. Presse´, "Mamba time series forecasting with uncertainty propagation," arXiv preprint arXiv:2503.10873, 2025.
19. J. Ye et al., "Empowering time series analysis with foundation models: A comprehensive survey," arXiv preprint arXiv:2405.02358, 2024.
20. F. Shi, X. Yin, K. Wang, W. Tu, Q. Sun, and H. Ning, "Large language models for time series analysis: Techniques, applications, and challenges," arXiv preprint arXiv:2506.11040, 2025.
21. C. Su, Z. Cai, Y. Tian, Z. Chang, Z. Zheng, and Y. Song, "Dif- fusion models for time series forecasting: A survey," arXiv preprint arXiv:2507.14507, 2025.
22. J. Li et al., "Deep learning-powered electrical brain signals analysis: Advancing neurological diagnostics," arXiv preprint arXiv:2502.17213, 2025.
23. J. P. Vieira, V. A. Bauler, R. K. Rosa, and D. Silva, "Towards a more realistic evaluation of machine learning models for bearing fault diagnosis," arXiv preprint arXiv:2509.22267, 2025.
24. R. R. Samani, A. Nunez, and B. De Schutter, "WaveletInception networks for drive-by vibration-based infrastructure health monitoring," arXiv preprint arXiv:2507.12969, 2025.
25. T. Dooney et al., "DeepExtractor: Time-domain reconstruction of signals and glitches in gravitational wave data with deep learning," arXiv preprint arXiv:2501.18423, 2025.
26. Y. Yan, J. Hou, Z. Song, and E. E. Kuruoglu, "Signal process- ing over time-varying graphs: A systematic review," arXiv preprint arXiv:2412.00462, 2024.
27. T. Inazumi, T. Washio, S. Shimizu, J. Suzuki, A. Yamamoto, and Y. Kawahara, "Causal discovery in a binary exclusive-or skew acyclic model: BExSAM," arXiv preprint arXiv:2509.01156, 2025.
28. Q. Zhang et al., "Time series analysis in frequency domain: A survey of open challenges, opportunities and benchmarks," arXiv preprint arXiv:2504.07099, 2025.
29. M. A. Ali and F. Dornaika, "Edge artificial intelligence: A systematic review of evolution, taxonomic frameworks, and future horizons," arXiv preprint arXiv:2510.01439, 2025.
30. J. Faouzi, "Deep learning for time series classification," in Deep Learning for Time Series Forecasting, Elsevier, 2021, ch. 8. [Online]. Available: https://doi.org/10.1016/B978-0-12-820491-7.00008-6
31. X. Yang, J. Li, and X. Jiang, "Research on information leakage in time series prediction based on empirical mode decomposition," Sci. Rep., vol. 14, art. no. 28362, 2024.
32. I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," Information, vol. 15, no. 9, p. 517, 2024.
33. Y. Li, Y. Lin, Y. Gao, and L. Khan, "Unified Spatio-Temporal Graph Neural Networks: Data-Driven Modeling for Social Science," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Padua, Italy, 2022, pp. 1–8.