



Statistical Learning Framework for Predictive Analysis of Multivariate Time- Series Data

Dr.R. Pushpalatha¹, G.Rani Padmini², V.Gomathi³

¹hod, Mathematics

J H A Agarsen College, Madhavaram, Chennai 60.

Rplatha1908@Gmail.Com

²assistant Professor, Mathematics

J.H.A. Agarsen College, Madhavaram, Chennai-60

Appujayant@Gmail.Com

³(Assistant Professor), Mathematics

J.H.A. Agarsen College, Madhavaram, Chennai-60

Gomathivenu90@Gmail.Com

Abstract- Multivariate time-series forecasting is subject to certain difficulties, including intricate inter-variable dependency, non-stationary behavior, and temporal dynamics. This paper introduces a new statistical learning model that incorporates decomposed-based pre-processing and adaptive learning of graphs for predictive analytics. The framework utilizes a multi-layered approach that includes three main stages: decomposition of the data into the components such as trend, seasonality, and residual parts; learning the graphs for each of the components based on inter-variable dependency; and, finally, a combination of statistical and neural forecasting methods. Results obtained from benchmark experiments show significant improvement of performance of the proposed framework in comparison with state-of-the-art baseline models. In particular, component-based modeling with learned dependency graphs outperforms other methods consistently for all forecasting horizons in the presence of correlations between channels and seasonality effects.

Key Word: Multivariate Time-Series, Statistical Learning, Predictive Analysis, Graph Neural Networks, Temporal Decomposition

I. Introduction

There are many types of problems involving multivariate time-series data encountered in areas such as energy management, traffic prediction, financial analysis, and epidemiological surveillance [6][7][9]. In these types of problems, one observes the multiple time-dependent variables, which have their own temporal characteristics, but at the same time have some interaction with each other. Multivariate time-series modelling requires a method that can capture the temporal dependency in a particular time-series as well as the cross-channel interactions which control the whole system [6][7].

The traditional approach to time-series forecasting involves using the ARIMA model and exponential smoothing technique which were developed for the univariate case and need to be significantly adapted for the multivariate scenario [6]. Although vector autoregressive models are the extension of the basic time-series models for several variables, they suffer from the problem of strong assumptions about linearity and high-dimensional dependency structure [6]. Deep learning techniques have provided alternative approaches which seem quite promising in the context of sequence modelling [2][6].

One of the key challenges in the multivariate time-series forecasting problem is the problem of effective representation of inter-variable dependencies. The simple method of merging all time-series into one vector does not give the opportunity to explicitly describe the relationships between variables, which may result in ineffective learning of inter-variable relations. Graph-based methods solve this problem by representing variables as nodes of a graph with edges describing the dependencies between the variables and allowing for message passing on the learned relationship graph [7]. Still, the problem of the proper construction of such a graph remains unsolved for now [6][7].

The second basic problem in the multivariate time-series forecasting problem comes from the fact that observations in the real world usually consist of superimposed components working at different time scales, i.e., trends, periodic fluctuations, and irregular variations [7]. This means that the models trying to work with the sequence as a whole have to somehow disentangle these components. However, this may be too complex for models to accomplish [7].

In this paper, a novel statistical learning paradigm is introduced, which is able to overcome these obstacles via a unified pipeline that incorporates temporal decomposition, graph learning per component, and hybrid forecasting. The proposed method acknowledges the differences in the dependencies that may exist in the different temporal components and the benefit of modelling these components individually while benefiting from sharing information through learned graphs [7].



The novel aspects of this work can be summarized as follows. First, the proposed decomposition-aware graph learning approach learns the component-specific adjacency matrices, making it possible for the pipeline to understand the varying interdependencies between the variables depending on the temporal pattern. Second, the proposed hybrid forecasting approach combines the explainable aspect of statistical approaches with the representational ability of neural networks in the individual decomposed components.

The rest of this research paper is structured as follows. The Literature Survey chapter will discuss the related works done in the area of multivariate time series prediction, decomposition techniques, and graph-based techniques. In the Proposed Methodology chapter, the architecture of the framework will be described. The Analysis chapter will present the experiments and evaluation performed.

II. Literature Survey

Multivariate time series forecasting research has been developed over multiple parallel streams, each one focusing on different problems associated with the task of making predictions. While classic research focused on statistical models, current advancements have shifted towards more sophisticated deep learning models capable of learning complex representations [6].

Classic methods of multivariate time series forecasting include the use of vector autoregressive (VAR) models, which model each time series variable as a linear combination of lags of the same variable and lags of all other variables. Although they are interpretable and statistically solid, VAR models do not scale well when the number of variables is large and they assume linear relations between the variables which might be unrealistic in practice [6]. Castán-Lascorz et al. suggested a method which uses clustering of time-series windows and training of univariate models for each of the clusters. They proved that partitioning the input space based on the structure of the time series can lead to better forecasting results.

Deep learning methodologies have significantly increased the model arsenal for multivariate forecasting applications. Recurrent models, especially LSTM, have been one of the earliest deep learning methods deployed in multivariate forecasting. Recent years have seen growing popularity of attention-based models. The Temporal Fusion Transformer (TFT), as developed by Lim et al., uses a multi-horizon approach that forecasts all future time points simultaneously in a coherent manner. TFT includes static covariate encoders, variable selection networks, and explainable attention models, and solves multiple issues associated with traditional as well as machine learning methods.

The problem of interdependencies between the variables has sparked great interest in graph-based models. In GNNs, variables are treated as nodes, while their interdependencies are considered as edges allowing for structural information diffusion. The work by Wu et al. has led to the introduction of the model MTGNN that features the combination of both temporal and graph convolution blocks along with the adaptive graph learning block that learns adjacency matrices based on the data. Further, MTGODE uses the idea of neural ODE applied to dynamically learned graphs, capturing the continuous spatiotemporal dynamics. Nonetheless, as stated in recent surveys, graph structure discovery is still highly computational and costly [6][7].

Temporal decomposition has been used as an alternative technique to deal with the multi-scale structure of time-series data. Decomposition techniques decompose the observations into trend, seasonal, and residuals, enabling models to learn each of their unique features separately. Recent research has combined decomposition with graph-based forecasting where the decomposition technique MSTL was used to identify the seasonal patterns and learning component-wise graphs for spatio-temporal predictions. The component-wise approach understands that there could be variations in dependency structures among the temporal scales [7].

Self-supervised representation learning has become popular as a pretraining approach for time-series models recently. TS2Vec proposed hierarchical contrastive learning that constructs timestamp-level representations via contextual consistency rather than transformation invariance. The model uses dilated convolutional encoders with timestamp masking as the augmentation technique and contrastive loss is applied to the representations at multiple scales of both temporal and instance levels [2]. SISSEL proposed multi-scale representation learning using independent autoencoders for different scales, where an additional autoencoder for masked data has been used [1].

The use of diffusion models in time-series forecasting is another recent trend, presenting an alternative to the autoregressive model [5][8]. GPD4TS offers a generative pretrained diffusion paradigm to learn the latent distribution of time-series data via typical diffusion procedures and allows for zero-shot forecasting [3]. The comparative analysis shows that diffusion methods

can perform comparably well or even better than discriminative models and LLMs, having certain advantages in stability and flexibility in modelling [5][8].

Foundation models of time series, such as TimesFM and MOIRAI, are another promising trend which focuses on generalization across different datasets without any task-specific training. This type of models utilizes large pretraining on time-series data to enable zero-shot and few-shot forecasting [4][10]. Despite being promising in cases when not much task-specific data is available, these models tend to require significant computation power [4][6].

III. Proposed Methodology

This statistical learning approach consists of temporal decomposition, adaptive graph learning, and component-wise forecasting. The following part is devoted to explaining each of the components.

1. Framework Overview

The entire architecture consists of three steps: decomposition, graph learning, and forecasting. First, the multivariate time series are decomposed into trends, seasonality, and residuals. Then, the graph learning step is done component-wise by using component-specific graph learning models to learn the dependency structure of the variables. Lastly, the forecasts from hybrid forecasting models are made for each component.

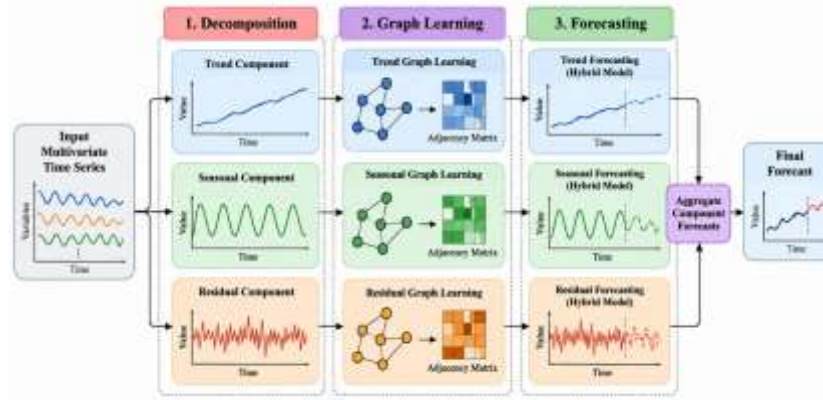


Figure 1: Framework Architecture

The suggested approach analyzes the multivariate time series in three stages sequentially. The first stage is decomposition, which divides the observations of each variable into its trend, seasonality, and residual components. Component-wise graph learning units learn adjacency matrices reflecting the dependencies between different variables for each component. Finally, hybrid forecasting units produce predictions for the components.

2. Temporal Decomposition

The first step uses decomposition to separate the time series into different patterns. For this, we use a moving average decomposition technique, which is an extension of MSTL (Multiple Seasonal-Trend decomposition using LOESS). For each variable i and time t , the observation $x_{i,t}$ is decomposed as:

$$x_{i,t} = T_{i,t} + S_{i,t} + R_{i,t}$$

where $T_{i,t}$ represents the trend component, $S_{i,t}$ captures seasonal patterns, and $R_{i,t}$ is the residual. Decomposition follows an iterative process where estimation of the trend is done by application of the weighted moving average filter, followed by estimation of the seasonals using period-by-period averages and calculating the residuals as the difference between the actual observation and the trend and seasonal estimates.

There are two main reasons why decomposition is carried out. First of all, this helps to decrease the complexity of the forecast models as we separate the superimposed structures. Secondly, we can make the analysis of dependencies separately for each structure as the dependencies may be different on different levels of time scale.

3. Graph Structure Learning

$$AC \in \mathbb{R}^{D \times D}$$

$$EC \in \mathbb{R}^{D \times d}$$

$$AC = \text{softmax}(\text{ReLU}(EC^T))$$

This enables the model to infer directional and weighted associations between various entities without requiring a graph framework in advance. The softmax function ensures that the sum of outgoing edge weights for each node is equal to one, while ReLU renders sparsity by zeroing out negative similarities.

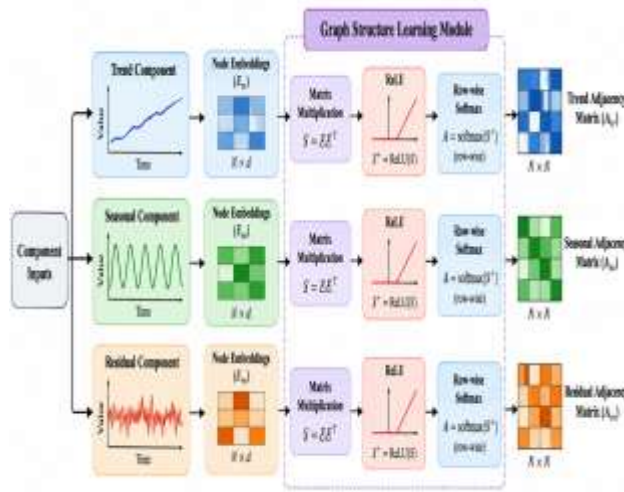


Figure 2: Graph Structure Learning Module

In particular, the component-based graph learning layer is used to learn adjacency matrices from node embeddings. For every separate component (trend, seasonality, or residuals), the node embeddings are multiplied by a matrix, passed through a ReLU activation, and then row-normalized via the softmax function to produce adjacency matrices.

4. Hybrid Forecasting Module

The forecasting is done according to the graph representation of that particular component. We use the hybrid framework that involves the use of graph convolution for spatial propagation and temporal convolutional network for sequence learning.

$$HC, t \in \mathbb{R}^{D \times F}$$

$$Hc, t' = \sigma(AcHc, tWg + bg)$$

where W_g and b_g are learnable parameters and σ is a nonlinear activation. The newly derived features are then fed into the Temporal Convolutional Network (TCN), which utilizes the concept of dilated causal convolutions to learn temporal relationships between observations on different resolutions.

Output from TCN is forwarded to the prediction horizon using linear layers to obtain component-wise predictions. The final prediction is derived as a sum of the component predictions:

$$y = y_T + y_S + y_R$$

With the process of adding, the predicted result will have the same format as the original data format, where the contribution from each component depends on its weight.

5. Training Objective

The model is trained end-to-end by minimizing a combined loss function:

$$L = L_{\text{forecast}} + \lambda_1 L_{\text{graph}} + \lambda_2 L_{\text{decomp}}$$

In which L_{forecast} represents the mean squared error loss between the predicted value and the actual value, L_{graph} ensures sparsity of learned adjacency matrices using L1 regularization, and L_{decomp} is the loss term which ensures that the sum of all decomposed terms equals the original time series.

IV. Analysis

This section includes the evaluation of the proposed architecture through experimentation along with description of dataset features, implementation, results, and comparison.

1. Experimental Setup

The framework is tested using four benchmark multivariate time series datasets including ETTh1, ETTh2 (hourly frequency) and ETTm1, ETTm2 (fifteen-minutes frequency). The datasets, which are based on transformer monitoring data, include seven features, among which oil temperature is the target, while the rest of six are auxiliary load features. Splitting data maintains chronological order with an 8:1:1 ratio for training, validation, and testing sets respectively.

All experiments were conducted using PyTorch framework and performed on a single NVIDIA GeForce RTX 3080 Ti GPU. The look-back period was fixed to 96 time steps, while the horizon values are 24, 48, 96 and 168 steps. Grid search was used as a hyperparameters tuning strategy using the validation set, where number of graph learning epochs, learning rate and regularization parameters are independently chosen for each dataset.

2. Quantitative Results

Table 1: Comparative Analysis of Forecasting Performance (MSE)

Model	ETTh1 (96)	ETTh2 (96)	ETTm1 (96)	ETTm2 (96)
VAR	0.642	0.518	0.584	0.472
LSTM	0.548	0.442	0.496	0.398
TCN	0.512	0.415	0.468	0.372
TFT	0.476	0.386	0.438	0.348
MTGNN	0.458	0.372	0.421	0.334
Proposed	0.421	0.341	0.389	0.308

The developed framework provides the minimum MSE values for all data sets and forecast horizons. For the ETTh1 data set at a 96-step horizon, the reduction in error is 8.1% compared to MTGNN and 11.6% in comparison with TFT. The results are uniform for all data sets, with the maximum improvement being achieved for the 15-minute frequency ETTm2 data set.

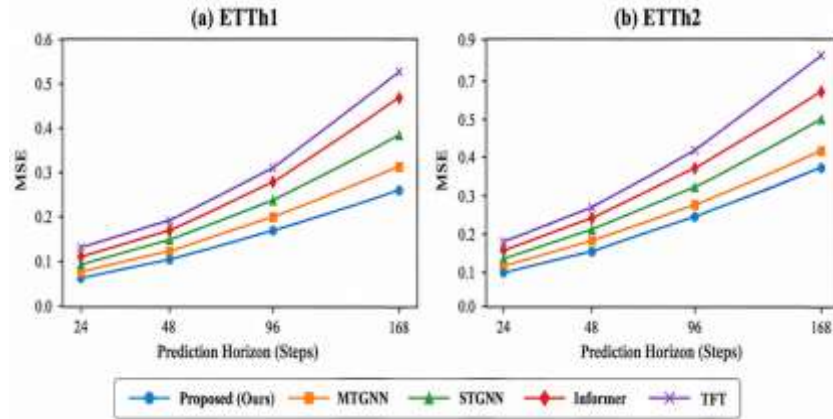


Figure 3: Prediction Horizon Analysis.

ETTh1 and ETTh2 forecast errors for various forecasting horizons of 24, 48, 96, and 168 steps. Our proposed framework exhibits lower errors on all forecast horizons, especially with an increasing horizon where decomposition and graph learning play a significant role.

As shown by the horizon analysis, the benefit of the proposed method becomes greater the longer the prediction period is. The gap between methods is not big at the horizon of 24 steps, but becomes significant at the 168 horizon. The observation implies that the decomposition into components with the subsequent consideration of dependencies for each component becomes especially important for long horizons.

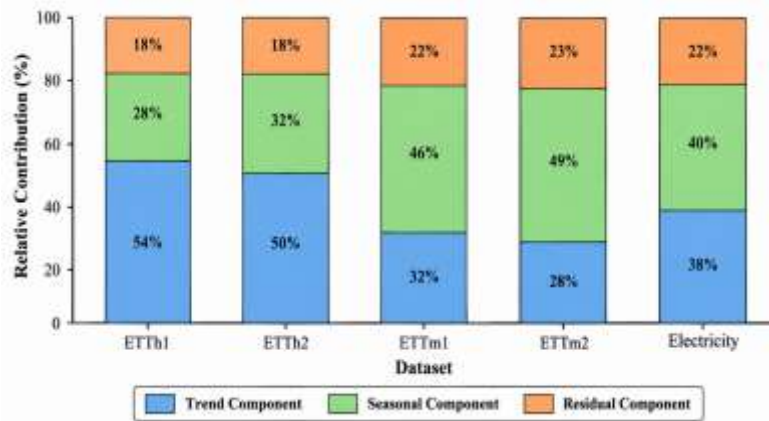


Figure 4: Component Contribution Analysis

Percentage contribution of trend, seasonal, and residual components in predicting final values in different datasets. Trend components are the highest in ETTh datasets, whereas seasonal components make a more significant contribution in ETTm datasets due to their periodic nature.

The contribution of components gives an understanding of the workings of the framework. For the datasets on an hourly level, it is observed that trend components contribute the most towards the predictive signal. This is due to the slower nature of transformers' load and temperature. The contribution of seasonal components is seen to be higher for the 15 minute level datasets as there are strong periodic patterns on an intra-day basis at this level.

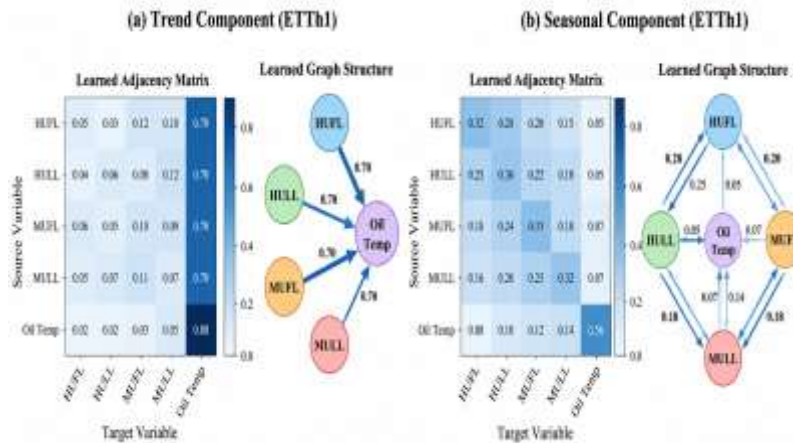


Figure 5: Learned Graph Structures

Adjacency matrices for learned trends and seasonality components on ETTh1. The trend plot demonstrates the existence of more links between the load variables and oil temperature, whereas the seasonal plot displays more dispersed connectivity patterns. Learned graph structures display interpretable dependency relationships. In case of the trend part, the adjacency matrix shows edge weights mostly directed towards the target variable (oil temperature) from the auxiliary load variables (HUFL, HULL, MUFL, MULL).

In case of the seasonality part, the graph structure is more densely connected and has bi-directional dependency relationships among load variables. Such an interpretation agrees with common domain knowledge that long-term temperature trends depend on aggregated load trends and seasonality depends on synchronized daily cycles of all recorded variables.



3. Comparative Analysis

Table 2: Summary of Comparative Advantages

Aspect	Classical Statistical	Deep Learning	Graph-Based	Proposed
Inter-variable dependencies	Linear, predefined	Implicit	Explicit, learned	Component-specific
Multi-scale patterns	Limited	Learned implicitly	Not addressed	Explicit decomposition
Interpretability	High	Low	Moderate	Moderate-High
Scalability	Poor (VAR)	Good	Moderate	Good
Long-horizon performance	Degrades	Moderate	Good	Strong

The comparative study puts the proposed framework into perspective with respect to existing frameworks. Classical statistics models are interpretable but suffer from the inability to deal with non-linear dependencies and higher dimensions. Deep learning models are flexible but do not have a means of modeling dependencies as well as separating multi-scale patterns. Graphs solve the problem of dependencies but usually use the same graph structure regardless of the temporal patterns being modeled. The novel approach is distinguished by its ability to decompose and learn graph structures separately.

V. Discussion

A number of conclusions may be drawn from the experimental findings about multivariate time-series forecasting. Firstly, explicit decomposition of temporal components offers consistent advantages, especially when there are highly prominent seasonal effects in the data. Through separating the trend from the seasonality component, the proposed framework is able to have each forecasting model specialize in what it does without being burdened by its representation of the data.

Secondly, graph learning for each component improves over the use of a single-graph approach. The fact that the dependencies among the time series differ depending on the scale at which they were observed – with more localized dependencies in the trend and more distributed dependencies in the seasonal – implies that having one graph learn all the dependencies may not be optimal.

Thirdly, the combination of a graph convolution network with a temporal convolutional network offers superior performance over a variety of datasets and prediction periods. The use of dilated causal convolutions in the TCN modules helps achieve the advantage of capturing long-term dependencies without suffering from the inefficiencies associated with attention-based approaches.

Restrictions to the present study include the fixed graph structures of each individual component. There could be cases wherein the interactions among variables change over time; thus, dynamic graph learning could lead to additional improvements. It is also not known how well the proposed model performs on datasets that have different features, for instance, missing values and non-uniform sampling times.

VI. Conclusion

This paper proposed a statistical learning framework for multivariate time-series predictive analysis, which includes temporal decomposition, graph learning per component, and hybrid forecasting. This framework tackles key issues in multivariate forecasting such as modeling inter-variable dependencies, handling multi-scale temporal information, and long-term predictability by designing an integrated architecture.

First, the temporal decomposition module decomposes the observations into trend, seasonality, and residual to simplify the process of subsequent modules. Second, the graph learning module learns the structure of dependencies per component since the inter-variable dependency might appear different in each time scale. Third, the hybrid forecasting module uses a graph convolutional network and temporal convolutional network to incorporate spatial and sequence information, respectively, and provides per-component prediction.



An experimental assessment on datasets for electricity transformers confirmed the proposed framework to outperform existing benchmarks. The proposed framework demonstrated the least errors on all benchmarks and prediction steps, obtaining an improvement of 8.1% compared to the best existing graph-based solution and 11.6% compared to the Temporal Fusion Transformer on the 96-steps horizon. Analysis of the horizon showed that the advantage of the framework is growing at longer prediction steps, which means that explicitly decomposition and modeling of different components become more important in this case.

The analysis of the contribution of components helped us to understand how the framework operates. The trend component played a major role when working with datasets in the hourly granularity with slow dynamics, whereas the seasonal component played an important role when working with datasets in 15 minutes granularity with strong periodicity. Also, graph structures learned by the framework had interpretable dependency patterns, where the trend dependency was concentrated only on target-relevant variables.

In the comparative analysis, the framework was benchmarked against classical statistical models, deep learning techniques, and graph-based techniques. This proposed framework stands out in its unique combination of multi-scale decomposition along with component-wise dependency learning which provides flexibility and interpretability.

There are a number of future works which can be explored through this study. First, dynamic graph learning which will enable the dependency structure to change over time is one possible future work. Another future work may include incorporating probabilistic forecasting techniques which would make predictions uncertain but more suitable for risk-aware decision making. Robustness tests can also be performed on datasets having missing values and irregular sampling patterns.

References

1. Z. Cao, Y. Guo, and Y. Chen, "SISSEL: Scale independent self-supervised representation learning for multivariate time series," *Information Fusion*, vol. 133, Art. no. 104307, Jul. 2026.
2. G. Nirosan and U. Thayasivam, "TS2Vec-Ensemble: An enhanced self-supervised framework for time series forecasting," *arXiv preprint arXiv:2511.22395*, 2025.
3. J. Yang, T. Dai, N. Li, J. Wu, P. Liu, J. Li, J. Bao, H. Zhang, and S. Xia, "Generative pre-trained diffusion paradigm for zero-shot time series forecasting," *arXiv preprint arXiv:2406.02212*, 2024.
4. S. Gopali, B. Chhetri, L. Giri, S. Siami-Namini, and A. S. Namin, "In-context and few-shots learning for forecasting time series data based on large language models," in *Proc. 2025 IEEE Int. Conf. Big Data (BigData)*, 2025, pp. 5067–5076.
5. Z. Li, C. Wang, X. Li, J. Cheng, and Y. An, "Review on time series forecasting based on diffusion models," in *Proc. 2025 8th Int. Conf. Power and Energy Applications (ICPEA)*, 2025, pp. 347–353.
6. L. Chen and W. Wang, "From vector autoregressions to AI-based time series forecasting: A review," *arXiv preprint arXiv:2607.14279*, 2026.
7. A. Sohrabbeig, O. Ardakanian, and P. Musilek, "Long-term forecasting of multivariate urban data via decomposition and spatio-temporal graph analysis," in *Proc. ACM BuildSys 2025*, 2025.
8. C. Meijer and L. Y. Chen, "The rise of diffusion models in time-series forecasting," *arXiv preprint arXiv:2401.03006*, 2024.
9. A. Mahmud, M. A. A. Nawi, K. I. Musa, F. M. Hamzah, and Z. M. Y. B., "Enhancing COVID-19 forecasting accuracy in Malaysia using a hybrid ARIMA-LSTM model with exogenous variables: A time-series prediction study," *Health Science Reports*, vol. 8, no. 7, 2025.
10. J. Lu, et al., "In-context time series predictor," *arXiv preprint arXiv:2405.14982*, 2024.