Graph Neural Networks in Intermittent Time-Series Forecasting

Mikhael Belmiro[1,a)], Finny Oktariani[1,b)]

[1]*Department of Computational Science, Institut Teknologi Bandung, Indonesia*
[2]*Combinatorial Mathematics Research Group, FMIPA, Institut Teknologi Bandung, Ganesha 10, Bandung 40132, Indonesia*

[a)] *belmiro533@gmail.com*

Abstract. Forecasting in the presence of intermittence is a challenging task. Several classic statistical approaches such as ARIMA is not directly applicable to such problem due to the stationary assumption ARIMA has. Fortunately, several methods such as Croston and ADIDA were devised to handle forecasting in the middle of intermittence. Generally, methods in forecasting intermittent time-series can be classified into two groups. The first group used inter-demand intervals and incorporated it in the forecasting model, and the second group used aggregation to obtain smoother time-series data. To the best of our knowledge, there is no model that tries to aggregate the effects of both methods. This paper tries to use fully convolutional neural networks (FCNN) with gating mechanisms to combine the methods present in each of the two groups while keeping the model efficient to train. Furthermore, this paper also tries to incorporate state-of-the-art model in graph neural networks in time-series forecasting to extract spatial information between time-series and incorporate it in the forecasting result, as research in graph neural networks for time-series forecasting is rising in popularity to address spatial dependence. Our model outperforms existing models which are prominent in the field of intermittent demand forecasting, and an ablation study is also done to address the effects of each part of the model. Our study shows that combining methods already established in intermittent demand-forecasting with state-of-the-art model in graph neural networks for time-series forecasting achieves better result in terms of metrics.

Keywords: Intermittent forecasting, fully convolutional neural networks, graph neural networks

INTRODUCTION

Demand forecasting is a crucial task, especially in businesses which rely on supply chain and management. For example, in online retails and e-commerce, businesses need to forecast the demand of their products to meet customers' needs. Failure in forecasting product demands could result in the loss of revenue and customer dissatisfaction. However, not all products are easily forecastable. Long-tail products, for example, may pose a challenge [1]. This kind of products is harder to forecast than its short-tail counterparts because it exhibits higher degree of intermittency in their demands or higher degree of variability in the number of products sold in each transaction period, while still contributing a considerable amount of revenue to the business [2]. Forecasting this kind of products is studied in a subproblem of demand forecasting, called intermittent demand forecasting.

Classical statistical models, such as ARIMA and its' family could not be used effectively in the middle of intermittency. This is due to the fact that ARIMA models require stationary assumption in which it is not applicable due to the number of zero values in the task of intermittent demand forecasting. Therefore, another method of solving this problem needs to be studied. These methods have been frequently discussed in studies regarding intermittent demand forecasting [1, 3, 4, 5, 6], therefore in this research paper several of these methods will be briefly discussed. In 1972, Croston introduced a method to solve this problem by using exponential moving average to forecast inter-demand periods and quantity of demand separately, which are then aggregated to produce a single point forecast. Croston method was further studied, in which SBA found that the original Croston method was biased and introduced a debiasing factor. This line of study of intermittent demand which uses inter-demand interval was continued by TSB in 2011, in which inter-demand interval was replaced by demand probability. Another line of study, which did not try to predict the inter-demand interval was introduced in 2011. This line of study aggregates the data points, make forecast, and then disaggregates the forecast to produce a single point forecast. An improvement of this work called IMAPA was then introduced in 2021, where instead of using only one aggregation level, IMAPA uses multiple aggregation levels. We note that solving intermittent demand forecasting problem has been predominantly dominated

I. H. Agustin (ed.), *Proceedings of the 2nd International Conference on Neural Networks and Machine Learning 2023 (ICNNML 2023)*, Advances in Intelligent Systems Research 183,
https://doi.org/10.2991/978-94-6463-445-7_13

been predominantly dominated by two kinds of models, one which uses inter-demand data and one which uses aggregate-disaggregate mechanisms. To the best of our knowledge, there are no works before that attempted to combine the effects of these two kinds of models. In addition to aggregating the effects of the two main modelling techniques, we also noted that these models are autoregressive in nature, which is computationally inefficient to forecast for longer horizons. Therefore, we experimented in using deep learning techniques to counter this problem. The problem of aggregating can be solved by using convolutional networks and the problem of creating inter-demand intervals can be solved by using gating mechanisms. Therefore, we use dilated causal convolutional networks which was used first in WaveNet [7] to extract temporal features and design a fully convolutional network for more efficient training. Our model is also designed to be nonautoregressive in nature.

In addition to using dilated causal convolutional networks, we also experimented in using graph neural networks to forecast intermittent demand, as products might be correlated by cross selling or upselling scheme, or more generally, product A might be demanded in higher quantity in a similar timeframe to product B. Previous works in graph neural networks for time-series forecasting was thoroughly studied in the field of traffic forecasting, in which models such as STGCN [8] and DCRNN [9] were first introduced in 2017. However, unlike these models which use spatial distance as proxy to spatial correlation, we directly learn spatial correlation from temporal patterns, in which series with similar temporal patterns will exhibit higher spatial correlations.

To summarize, the contribution of this study is threefold, (1) we attempt to combine two predominant methods in solving intermittent demand forecasting, (2) we designed a neural network in which is nonautoregressive for intermittent demand forecasting, and (3) we attempt to bridge the discrepancy in graph neural networks for time-series forecasting and intermittent demand forecasting by incorporating techniques present in the graph neural networks for time-series forecasting in intermittent demand forecasting.

The structure of this paper is as follows. The first section covers the introduction to the paper. The second section covers some preliminaries regarding dilated causal convolutional layer and STGM [10]. The third section covers the model architecture that we use in our paper. The fourth section covers the experiments that we do. The fifth section covers conclusion and future works regarding intermittent demand forecasting with graph neural networks.

METHOD

Problem Statement

A standard multivariate time-series forecasting problem setup can be defined as follows. Define $X \in R^{T \times n \times c}$ to be a multivariate time-series data where T is the number of historical horizons, n the number of single time-series, and c to be the number of channels in the time-series. We want to predict $Y \in R^{P \times n \times c}$ where P is the number of prediction horizons. We note that $x_j^{(i)}$ and $y_j^{(i)}$ to be the value of a time-series j at time i for historical and future time-series. Our model can be written as a function F that minimizes a loss function $L\left(Y, \hat{Y}\right)$ is. We choose our loss function L to be MSE loss, that is $L\left(Y, \hat{Y}\right) = \frac{1}{nP} \sum_{i=1} \sum_{j=i} \left(y_j^{(i)} - \hat{y}_j^{(i)}\right)^2$. Because we are only dealing with historical time-series data with no additional exogenous variables, here $c = 1$ and further notations will be written as if $c = 1$.

Dilated Causal Convolutional Layer

Dilated causal convolutional layer was used to extract temporal information across temporal dimension by using only convolutional layers. Dilated causal convolutional layer was first used to generate raw audio waveform from previous audio waveform in a regressive manner. The term causal from dilated causal convolutional layer was noted so that future values can only be generated using previous values, that is $x_j^{(i)} = f(x_j^{(i-1)}, x_j^{(i-2)}, \ldots, x_j^{(1)})$. Therefore, dilated causal convolutional layer is suitable for time-series forecasting and is more efficient than RNN and its derivations due to its parallelizability.

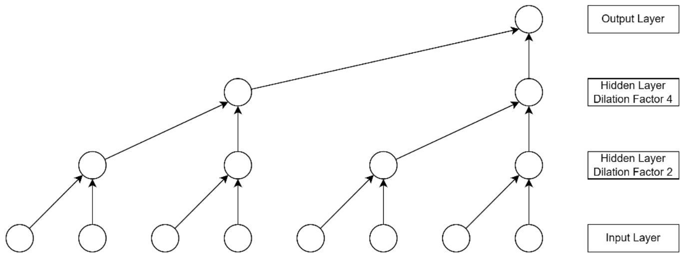

FIGURE 1. Visualization of dilated causal convolutional layer taken from [9]

Dilated causal convolutional layer is constructed as a stack of convolutional layers connected via residual gates and skip connections as illustrated in Figure 1. and Figure 2. Dilation controls the receptive field of the convolutional kernels to achieve bigger receptive fields while keeping the operations to a minimum. However, unlike WaveNet, we use the final layer's skip output instead of aggregating skip outputs of different layers.

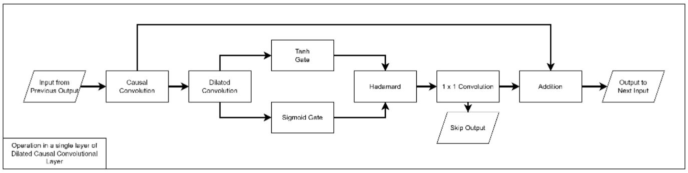

FIGURE 2. An overview of the residual connections in a single layer of dilated causal convolutional layer as taken from [9]

STGM

As mentioned previously, our GNN model relies on temporal extraction to capture similar patterns in each time-series to construct spatial correlations. Our GNN mimics a lot of STGM architecture, in which STGM was one of the first models which utilized temporal data to construct spatial correlations. As shown in Figure 3., in addition to static graph learning methods [13], STGM uses transformers [14] to extract temporal patterns from the data, an approach which was quite ubiquitous in time-series forecasting [15, 16, 17, 18], and also a similarity estimator that estimates the cross-correlation of historical to future series. The spatio-temporal graph attention (STGA) block in STGM creates query (Q), key (K), and value (V) pairs from X by using convolutions across the time dimension. Q and K for each of the convolutions are then passed through an operation $A = Softmax\left(QK^T\right)$ to create the spatial correlations matrix A. A is then enriched with static embeddings from the graph learning method and similarity estimator matrix from the similarity estimator to create a final representation of spatial correlations matrix $\overline{A}$. V is then fed to the spatial correlations matrix A through matrix multiplication $V\overline{A}$ to produce an output for a single layer of STGA. Multiple outputs of STGA are passed into exponential linear unit layers (ELU) and aggregated using a linear mixer.

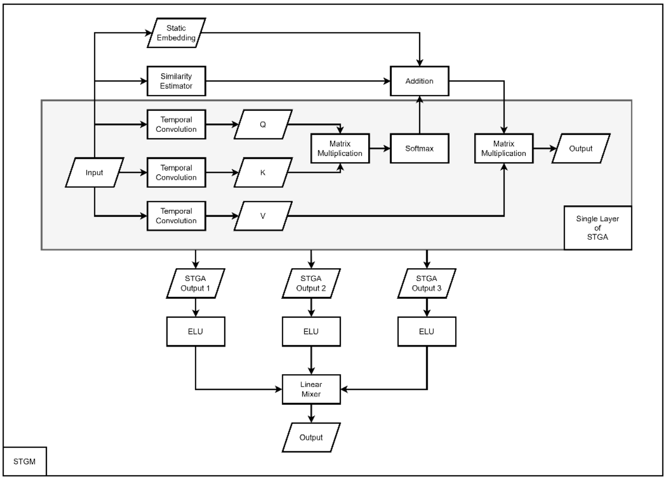

FIGURE. 3. STGM architecture as taken from [12]

Model Architecture

The model consists of three main blocks in which the data will be fed to, a gated fully convolutional neural network (FCNN) block, regression FCNN block, and an STGM block, as illustrated in Fig. 4. More details will be explained in Section 3.1 and Section 3.2.

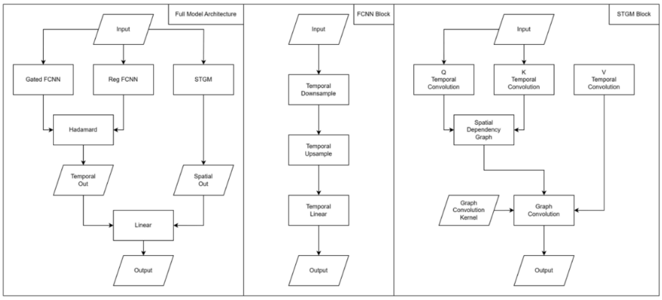

FIGURE 4. (from left to right) Full model architecture, FCNN block, STGM block

FCNN Block

The Gated FCNN block and Reg FCNN block are two blocks that can be summarized by the FCNN block. The FCNN block consists of a temporal downsample dilated causal convolutional layer and a temporal upsample dilated causal convolutional layer. As seen in aggregate-disaggregate method, the temporal downsample dilated causal convolutional layer aggregates different timestamps of the input historical and temporal upsample dilated convolutional layer disaggregates aggregated values from the temporal downsample dilated causal convolutional layer. The final temporal linear transforms the final disaggregated value to match the prediction horizon. Mathematically, the operations happening in the block can be written as follows.

$$H^{(1)} = TempDownDConv(X),\ H^{(1)} \in R^{T_{out} \times n \times c_{out}}$$
$$H^{(2)} = TempUpDConv\left(H^{(1)}\right),\ H^{(2)} \in R^{T \times n \times c_{out}}$$
$$\hat{Y}^{Reg\ FCNN} = Linear\left(H^{(2)}\right),\ \hat{Y}^{Reg\ FCNN} \in R^{P \times n}$$

Specifically for Gated FCNN, the output of the linear layer is then passed into a sigmoid activation function to mimic approximating the probability of demand occurring and a timestep. Mathematically, the formulation can be written as follows

$$H^{(1)} = TempDownDConv(X),\ H^{(1)} \in R^{T_{out} \times n \times c_{out}}$$
$$H^{(2)} = TempUpDConv\left(H^{(1)}\right),\ H^{(2)} \in R^{T \times n \times c_{out}}$$
$$H^{(3)} = Linear\left(H^{(2)}\right),\ H^{(3)} \in R^{P \times n}$$
$$\hat{Y}^{Gated\ FCNN} = Sigmoid\left(H^{(4)}\right),\ \hat{Y}^{Gated\ FCNN} \in R^{P \times n}$$

Finally, the output of both of these FCNN blocks is aggregated via the Hadamard function, which can be mathematically written as follows.

$$\hat{Y}^{FCNN} = \hat{Y}^{Gated\ FCNN} \odot \hat{Y}^{Reg\ FCNN},\ \hat{Y}^{FCNN} \in R^{P \times n}$$

The output of the FCNN blocks will be aggregated with the output from STGM block.

STGM Block

As mentioned before, mimics a lot from the original STGM architecture. Instead of using the whole model, we will be only using STGA block to create spatial correlations A and pass A through a graph convolutional operator as done in [13]. Mathematically, the operations can be written as follows.

$$Q,\ K,\ V = Att(X)$$
$$A = Softmax(QK^{T})$$
$$\hat{Y}^{STGM} = \Theta(I + A)V$$

Here, the spatial correlations matrix A acts as a parameterized replacement of eigenvectors of the Laplacian of a graph and it is used to transform V to its frequency domain to be convolved in the frequency domain.

Final Linear

The final linear is used to combine output from the FCNN block and the STGM block. Mathematically, it can be written as follows, where [...||...] denotes the concatenation operation.

$$\hat{Y} = Linear\left(\left[\hat{Y}^{FCNN} || \hat{Y}^{STGM}\right]\right),\ \hat{Y}^{FCNN} \in R^{P \times n}$$

Results and Discussion

Dataset

We experimented our model on 3 real world datasets.

- Car Parts

Car parts data comprises of 51 months monthly data of 2,674 timeseries. The data is available online in [31].

- RAF

RAF dataset comprises of 84 months monthly data of 5,000 aerospace parts demand. This data is taken from Royal Air Force. The data is available online in [32].

- Auto

Auto dataset comprises of 24 months monthly data of 3,000 auto parts demand. The data is available online in [32].

The summary of the dataset can be seen at Table 1. *ADI* (Average Demand Interval) signifies how many nonzero demands are present compared to the length of the time-series, lower value means that nonzero demand appears more frequent. CV^2 (Squared Coefficient of Variation) measures the variability in demand, lower value means that the timeseries has lower variability. The classification of smooth, intermittent, lumpy, and erratic follows SBA classification [4, 20] of time-series. Based on the table, we can see that each dataset has considerable amounts of series other than smooth series, which shows that the dataset is suitable for this particular problem. We will be forecasting to 6 future horizons by using past 6 observational data.

TABLE 1. Summary of the dataset

Dataset	Auto	RAF	Car Parts
Mean ADI	1.32	11.14	7.49
Mean CV^2	0.41	0.64	0.27
Smooth Pct (%)	41.37	0	0
Intermittent Pct (%)	35.8	58.1	86.57
Lumpy Pct (%)	10.23	41.9	13.43
Erratic Pct (%)	12.6	0	0

Result Comparison

To compare the results, we use 2 frequently used metrics in intermittent time-series forecasting, that is $RMSE$ and $RMSSE$. Below is the mathematical formulation for both $RMSE$ and $RMSSE$ for each of the series. Loss values for each series are then averaged across each series.

$$RMSE(j) = \sqrt{\sum_{i=1}^{P} \left(\hat{y}_j^{(i)} - y_j^{(i)}\right)^2}$$

$$RMSSE(j) = \sqrt{\frac{1}{P} \frac{\sum_{i=T+1}^{T+P} \left(\hat{y}_j^{(i)} - y_j^{(i)}\right)^2}{\frac{1}{n-1}\sum_{j=2}^{T} \left(y_j^{(i)} - y_j^{(i)}\right)^2}}$$

$RMSSE$ is a variation of the $MASE$ that has gained popularity in recent years. $MASE$ is a suitable measure in measuring model performance in intermittent demand forecasting since it measures the relative errors of each of the time-series while also being robust to the frequent number of zero demands, since it scales the mean squared error in the numerator by comparing it with in-sample data treated as random walk forecast, unlike $MAPE$ which will be undefined for data with zero demands.

From Table 2., we can see that our model (ITNGF) outperforms the other model in terms of both $RMSE$ and $RMSSE$. It is also notable that the models based on aggregate-disaggregate method achieves better result in general, as opposed to models that predicts the inter-demand arrival time. However, combining the two methods using deep learning in addition to graph neural networks improves the performance of the model.

TABLE 2. Model comparison across all datasets

Datasets	Model	RMSE	RMSSE
Auto	Croston	8.084991	1.555839
	SBA	8.800440	1.597474
	TSB	8.448380	1.522256

Datasets	Model	RMSE	RMSSE
	ADIDA	8.264835	1.492977
	IMAPA	8.249843	1.486004
	ITNGF	**8.095933**	**1.444035**
RAF	Croston	20.698347	1.482354
	SBA	20.308754	1.445033
	TSB	18.105827	1.167171
	ADIDA	17.377874	1.163829
	IMAPA	17.128208	1.163118
	ITNGF	**16.686794**	**1.156339**
Car Parts	Croston	1.519452	0.794615
	SBA	1.471653	0.748317
	TSB	1.106884	0.452554
	ADIDA	1.105914	0.450473
	IMAPA	1.083539	0.446223
	ITNGF	**1.045698**	**0.432099**

Diving deeper into the RAF dataset, as shown in Table 3., we can see that our model outperforms the other models in terms of forecasting both intermittent time-series and lumpy time-series.

TABLE 3. Model comparison in RAF disaggregated by time-series classification

Classification of TS	Model	RMSE	RMSSE
Intermittent	Croston	26.100779	1.816995
	SBA	25.341681	1.711956
	TSB	20.826457	1.118378
	ADIDA	19.232345	0.969510
	IMAPA	18.735551	0.912418
	ITNGF	17.841560	0.815757
Lumpy	Croston	16.374712	18.825332
	SBA	16.333515	18.748421
	TSB	16.145245	18.398667
	ADIDA	16.091846	18.298634
	IMAPA	16.025890	18.178220
	ITNGF	15.913298	17.951510

Ablation Study

In this ablation study, we only use RAF dataset since it is the only dataset with only intermittent and lumpy time-series, which are the subject of interest in this study. From Table 4., we can see that our model achieves the best performance. However, it is worth noting that our model exhibits a significant drop in performance after removing the STGM block from our model, while a comparable result is shown by using only the STGM block. This shows that state-of-the-art model in time-series forecasting is also sufficient in forecasting intermittent demand since it also extracts temporal information from the data. This is also supported by the fact that STGM achieves better performance than directly forecasting demand values by using only Reg FCNN. However, the presence of Gated FCNN in our model is necessary to control the effect of Reg FCNN to STGM for the aggregation to be effective.

TABLE 4. Ablation study of ITNGF model

Model	RMSE	RMSSE
ITNGF	16.686794	1.156339
ITNGF No STGM	16.904516	1.262679
ITNGF No Gated FCNN	16.888088	1.193248

ITNGF No STGM & Gated FCNN (Reg FCNN Only)	16.895565	1.235920
STGM Only	16.697180	1.159825

Conclusion

Forecasting in the presence of intermittency is a challenging task because of the prominent presence of zero values and the high variability of data. However, it is also important as businesses in the retail or e-commerce industry also experience periods of no demand, in which intermittent demand forecasting could be an integral part of the business as an effort to reduce the cost of having too much supply of a specific product and to understand customer behavior in the future. However, we found that the growth in the research of intermittent demand forecasting to be lagging as opposed to its more general time-series forecasting counterpart.

In this work, we devised a model which combines methods which are commonly used in intermittent demand forecasting along with graph neural networks which is gaining popularity in the field of time-series forecasting. Our model outperforms prominent models in intermittent demand forecasting and also achieves the best result when focusing only on time-series which are classified as intermittent and lumpy. Our work also is the first to make connections to graph neural networks in time-series forecasting by using parts of a state-of-the-art model in graph neural networks for time-series forecasting, which opens up possibilities of utilizing graph neural networks alongside already established methods in intermittent demand forecasting to improve models in the context of intermittent demand forecasting. We believe that our work is also the first to incorporate state-of-the-art time-series forecasting method in intermittent demand forecasting and shows that using the state-of-the-art time-series forecasting method is sufficient in solving intermittent demand forecasting. We also believe that segregating time-series into different classification, as done by SBA, could provide better analysis on how models perform differently in different contexts of series, which is a good direction for researchers to also focus on onwards.

Acknowledgment

First and foremost, praises to God for the blessings that He has given to us throughout this research. We would also like to show our deepest appreciation to PENELITIAN, PENGABDIAN KEPADA MASYARAKAT, DAN INOVASI (PPMI) Kelompok Keilmuan Matematika Kombinatorika FMIPA ITB 2022 Research Grant since this work cannot be done without the funding of the grant. We would also like show our gratefulness to the committee of The 2nd International Conference on Neural Networks (ICONNSMAL) 2023 for organizing the conference and giving clear instructions on how to structure this paper.

References

1. Türkmen. A. C., Januschowski, T., Wang, Y., Cemgil, A.T.: Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. PLoS ONE 16(11): e0259764 (2021).
2. Investopedia: Long Tail: Definition as a Business Strategy and How It Works, https://www.investopedia.com/terms/l/long-tail.asp#:~:text=The%20long%20tail%20of%20distribution,reduced%20marketing%20and%20distribution%20costs., last asccessed 2024/01/03.
3. R., K., Kayathwal, K., Dhama, G., Arora, A.: A Survey on Classical and Deep Learning based Intermittent Time Series Forecasting Methods. International Joint Conference on Neural Networks (2021).
4. Yilmaz, T., Yapar, G., Yavuz, I.: Time Series Forecasting of Intermittent Demand by using ATA Method. Mugla Journal of Science and Technology (2019).
5. Azizi, F., Wibowo, W. C.: Intermittent Demand Forecasting Using LSTM With Single and Multiple Aggregation. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 6(5), 855 – 859 (2022).
6. Kiefer, D., Grimm, F., Bauer, M., van Dinther, C.: Demand Forecasting Intermittent and Lumpy Time Series: Comparing Statistical, Machine Learning and Deep Learning Methods (2021)
7. Oord, et. al.: WaveNet: A Generative Model for Raw Audio. arXiv preprint arXiv: 1609.03499 (2016).
8. Yu, et. al.: Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (2018).

9. Li, et. al.: Graph Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. arXiv preprint arXiv: 1707.01926 (2018).
10. Lablack, M., Shen, Y.: Spatio-temporal graph mixformer for traffic forecasting. Expert Systems with Applications. Vol. 228 (2023).
11. Wu, et. al.: Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. arXiv preprint arXiv: 2005.11650 (2020).
12. Vaswani, et. al.: Attention Is All You Need. arXiv preprint arXiv:1706.03762 (2017).
13. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106-11115.
14. Zhou, et. al.: FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. Proceedings of the 39th International Conference on Machine Learning, in Proceedings of Machine Learning Research 162:27268-27286 (2022).
15. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. arXiv preprint arXiv: 2106.13008 (2021).
16. Nie, Y., Nguyen, N., Sinthong, P., Kalagnanam, J.: A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. arXiv preprint arXiv: 2211.14730 (2022).
17. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are Transformers Effective for Time Series Forecasting? arXiv preprint arXiv:2205.13504 (2022).
18. Rangapuram, S.S., Werner, L.D., Benidis, K., Mercado, P., Gasthaus, J., Januschowski, T.: End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series. Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139:8832-8843 (2021).
19. Kourentzes, N.: Intermittent demand forecasts with neural networks. International Journal of Production Economics 143(1), 198-206 (2013).
20. [Spiliotis, et. al.: Comparison of statistical and machine learning methods for daily SKU demand forecasting. Oper Res Int J 22, 3037–3061 (2022).
21. F. Lolli, R. Gamberini, A. Regattieri, E. Balugani, T. Gatos, S. Gucci: Single-hidden layer neural networks for forecasting intermittent demand. International Journal of Production Economics 183, 116-128 (2017).
22. Luana, R., Fernando, G., Alejandro, R.: Gated Graph Recurrent Neural Networks. IEEE Transactions on Signal Processing 68, 6303-6318 (2020).
23. Wu, et. al.: Graph WaveNet for Deep Spatial-Temporal Graph Modeling. arXiv preprint arXiv: 1906.00121 (2019).
24. Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H.: Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 922-929 (2019).
25. Rob J. Hyndman R Package containing data from the book "Exponential smoothing: a state space approach", https://github.com/robjhyndman/expsmooth, last accessed 2023/11/17.
26. Alexandrov, A. and Benidis, K. and Bohlke-Schneider, M. and Flunkert, V. and Gasthaus, J. and Januschowski, T. and Maddix, D. C. and Rangapuram, S. and Salinas, D. and Schulz, J. and Stella, L. and Türkmen, A. C. and Wang, Y.: GluonTS: Probabilistic Time Series Modeling in Python. arXiv preprint arXiv:1906.05264 (2019).