Time Series Forecasting

Soroosh Shalileh

June 1, 2025

Abstract

...

1 Problem Formulation

The objective of the current work is to forecast the **total consumption** for a given category of merchants over time using historical transaction data.

Let $t \in \{1, 2, \ldots, T\}$ denote discrete time steps, which may correspond to days, weeks, or months. Let $\mathcal{M}_c = \{m_1, m_2, \ldots, m_N\}$ represent the set of merchants in category c, where N is the number of merchants. For each merchant $m \in \mathcal{M}_c$, let $x_{m,t} \in \mathbb{R}$ denote the transaction volume or total spend at time t.

We define the vector of merchant-level consumption at time t as

$$X_t = \{x_{m,t} \mid m \in \mathcal{M}_c\} \in \mathbb{R}^N,$$

and the total category-level consumption as

$$y_t = \sum_{m \in \mathcal{M}_c} x_{m,t}.$$

Let $\mathcal{H}_t = \{X_{t-k}, X_{t-k+1}, \ldots, X_t\}$ denote the historical sequence of merchant-level transaction data for the past $k+1$ time steps. The forecasting objective is to learn a function f_θ, parameterized by (deep learning) model parameters θ, such that

$$\hat{y}_{t+1} = f_\theta(\mathcal{H}_t),$$

This formulation supports several forecasting variants. One can perform one-step forecasting (predicting y_{t+1}), multi-step forecasting (predicting $y_{t+1}, y_{t+2}, \ldots, y_{t+H}$), or even jointly forecast the category total and the individual merchant-level consumptions.

2 Deep Learning Models for Forecasting

We treat the problem as a sequence regression task, where the input is a time-series sequence of merchant-level transaction vectors, and the output is the future total consumption value. Several model families are suitable for this task.

2.1 Input Representation

At each time step t, the input feature vector is

$$X_t = [x_{m_1,t}, x_{m_2,t}, \ldots, x_{m_N,t}] \in \mathbb{R}^N$$

This vector may be optionally augmented with additional contextual features such as time-of-day, day-of-week, holiday indicators, and merchant embeddings if available. The model receives a sequence $\mathcal{H}_t = \{X_{t-k}, \ldots, X_t\}$ of such vectors as input. [1]

[1] Each vector $X_t \in \mathbb{R}^N$ represents the transaction values across N merchant-category combinations (e.g., spending at each merchant or category at time t). To prepare this data for (deep learning) models, we stack these vectors along the temporal axis to form a matrix $\mathcal{H}_t \in \mathbb{R}^{(k+1)\times N}$, where each row corresponds to the transactions at a specific time step. This structure allows models such as LSTMs, TCNs, and Transformers to process the sequence as a multivariate time series input.

2.2 Model Architectures

Fully Connected Network (MLP): A simple baseline approach involves flattening the entire input history and passing it through a multi-layer perceptron (MLP). That is, the input is reshaped into a vector in $\mathbb{R}^{(k+1)\times N}$, which is then fed into dense layers to predict y_{t+1}. While MLPs are not inherently designed for sequential data, they have been effectively applied in time series forecasting tasks. For instance, the study by Zhang et al. (2018) demonstrated the use of MLPs in forecasting electricity consumption, highlighting their capability in capturing complex nonlinear relationships in time series data [1].

Recurrent Neural Networks (LSTM/GRU): Recurrent models process the temporal sequence $\{X_{t-k}, \ldots, X_t\}$ step by step, maintaining a hidden state that captures temporal dependencies. These models are well-suited for sequential data and can model long-term patterns. Long Short-Term Memory (LSTM) networks, in particular, have been widely used in time series forecasting. For example, the work by Elsworth and Güttel (2020) applied LSTM networks for time series forecasting using a symbolic approach, demonstrating their effectiveness in capturing temporal dynamics [2]. A simplified alternative to LSTM is the Gated Recurrent Unit (GRU), which merges the forget and input gates into a single update gate and combines the hidden state and cell state, resulting in a more computationally efficient model. GRUs have shown competitive performance across various sequential prediction tasks [3].

Temporal Convolutional Networks (TCNs): TCNs use one-dimensional convolutions over time to capture local and hierarchical temporal patterns. They are efficient and effective for long sequences and avoid the vanishing gradient problems of RNNs. The study by Bai et al. (2018) provided a comprehensive empirical evaluation of TCNs, showing their superior performance over RNNs in various sequence modeling tasks, including time series forecasting [4].

Convolutional LSTM (ConvLSTM): ConvLSTM extends traditional LSTM networks by incorporating convolution operations into both the input-to-state and state-to-state transitions. This architecture is particularly effective for modeling spatiotemporal data, where inputs exhibit both spatial and temporal dependencies. In the context of merchant transaction data, if merchants are organized based on geographic or semantic relationships, ConvLSTM can capture localized patterns over time. For instance, Shi et al. (2015) introduced ConvLSTM for precipitation nowcasting, demonstrating its superiority over traditional LSTM models in capturing spatiotemporal correlations [5]. Further applications, such as in the ED-ConvLSTM model for forecasting global ionospheric total electron content, have showcased ConvLSTM's capability in handling complex spatiotemporal forecasting tasks [6].

Hybrid TCN-LSTM Models: Combining Temporal Convolutional Networks (TCNs) with LSTM architectures leverages the strengths of both models—TCNs excel at capturing local temporal patterns through dilated causal convolutions [2], while LSTMs are adept at modeling long-term dependencies. In a hybrid TCN-LSTM model, the TCN component first processes the input sequence to extract high-level temporal features, which are then fed into the LSTM to capture sequential dependencies over longer horizons. This architecture has been effectively applied in various forecasting tasks. For example, a hybrid model combining TCN and LSTM demonstrated improved performance in wind power prediction by effectively capturing both short-term fluctuations and long-term trends [7]. Similarly, in the realm of electric load forecasting, integrating TCN with LSTM has led to enhanced predictive accuracy, highlighting the synergy between these architectures [8].

Transformer Encoder: Transformer-based models can be used to process the input sequence as a set of time-embedded transaction vectors. These models use self-attention mechanisms to learn dependencies across time steps. The sequence is encoded, and a learned summary representation is passed through a feedforward network to predict y_{t+1}. The Transformer architecture, introduced by Vaswani et al. (2017), has been adapted for time series forecasting in various studies. For instance, the PatchTST model proposed

[2]Here "causality" refers to a convolution operation that does not use any future time steps when predicting the current output.

by Nie et al. (2022) demonstrated the effectiveness of Transformers in long-term time series forecasting by segmenting time series into patches and applying channel-independent attention mechanisms [10]. Other adaptations include the Informer model, which improves efficiency via a ProbSparse attention mechanism [3] tailored for long-sequence forecasting [11], and Autoformer, which incorporates a decomposition block to explicitly model seasonal and trend components [12]. These improvements make Transformer-based models particularly suitable for multiscale and long-horizon forecasting tasks.

2.3 Loss Function

Assuming the model outputs a scalar prediction $\hat{y}_{t+1}$, the objective is to minimize the mean squared error (MSE) over all training sequences:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{T-k} \sum_{t=k}^{T-1} \left(\hat{y}_{t+1} - y_{t+1}\right)^2 .$$

Other loss functions such as mean absolute error (MAE) or Huber loss can be employed depending on the characteristics of the target distribution. Additionally, multi-task losses can be introduced if merchant-level forecasts are also required.

3 Experimental results

References

[1] Zhang, G., Eddy Patuwo, B., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1), 35–62. https://doi.org/10.1016/S0169-2070(97)00044-7

[2] Elsworth, S., & Güttel, S. (2020). Time Series Forecasting Using LSTM Networks: A Symbolic Approach. *arXiv preprint arXiv:2003.05672*. https://arxiv.org/abs/2003.05672

[3] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*. https://arxiv.org/abs/1406.1078

[4] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*. https://arxiv.org/abs/1803.01271

[5] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-C. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Advances in Neural Information Processing Systems*, 28. https://arxiv.org/abs/1506.04214

[6] Xia, X., Zhang, Y., Wang, H., & Liu, Y. (2022). ED-ConvLSTM: A Novel Global Ionospheric Total Electron Content Medium-Term Forecast Model. *Space Weather*, 20(3), e2021SW002959. https://doi.org/10.1029/2021SW002959

[7] Wang, Y., Zhang, L., & Liu, H. (2024). A hybrid deep learning model based on parallel architecture TCN-LSTM with Savitzky-Golay filter for wind power prediction. *Energy*, 290, 120000.

[8] Rao, A. A., Rao, P. M., & Kumar, D. V. (2024). Hybrid TCN-Based Bi-GRU-LSTM for Enhanced Long-Term Electric Load Forecasting. *Journal of Electrical Systems*, 20(3). https://journal.esrgroups.org/jes/article/view/7486

[3] The ProbSparse attention mechanism, reduces the quadratic complexity of full self-attention by selecting only the top-u queries with the largest sparsity scores. This enables the model to focus on the most informative parts of long input sequences while maintaining computational efficiency.

Table 1: Evaluation Metrics (MAE, MAPE, R^2) for Each Model Across Consumption Categories

Model	**Category**	**MAE**	**MAPE (%)**	R^2
MLP	Food			
	Non-food			
	Cafe			
	Service			
LSTM	Food			
	Non-food			
	Cafe			
	Service			
GRU	Food			
	Non-food			
	Cafe			
	Service			
TCN	Food			
	Non-food			
	Cafe			
	Service			
Hybrid TCN-LSTM	Food			
	Non-food			
	Cafe			
	Service			
Transformer	Food			
	Non-food			
	Cafe			
	Service			
PatchTST	Food			
	Non-food			
	Cafe			
	Service			
Informer	Food			
	Non-food			
	Cafe			
	Service			
Autoformer	Food			
	Non-food			
	Cafe			
	Service			

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. https://arxiv.org/abs/1706.03762

[10] Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2022). A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *arXiv preprint arXiv:2211.14730*. https://arxiv.org/abs/2211.14730

[11] 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. https://arxiv.org/abs/2012.07436

[12] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 22419–22430. https://arxiv.org/abs/2106.13008