Transparent Time Series Forecasting with Static and Non-Static Features

Silas Ruhrberg Estévez
University of Cambridge
sr933@cam.ac.uk

Abstract

Time-series forecasting is essential in many real-world applications, such as medicine, scientific discovery, and finance. While traditional models like RNNs have achieved high accuracy across various tasks, they often lack transparency, making their predictions challenging to interpret and prone to biases. Recent advancements have introduced transparent forecasting methods using B-Spline piecewise approximations, which enhance interpretability. However, these models are constrained to predicting time-series data based solely on static features, limiting their applicability in dynamic, real-world scenarios. To overcome this limitation, we propose a novel approach that builds on the strengths of B-Splines for transparent forecasting, enabling the integration of both static and dynamic (non-static) features by adapting the spline coefficient prediction process. The accuracy of the approach is demonstrated on three synthetic datasets and a real-world intensive care unit dataset.

1 Introduction

Timeseries forecasting importance

Time series are a type of sequential data that capture how the properties of a system evolve over time [1]. Accurately predicting the future behavior of such systems has widespread applications across various domains, including meteorology [2, 3], medicine [4, 5], and finance [6]. Over the years, a range of models have been developed for time series forecasting, spanning traditional signal processing techniques [7] to advanced deep learning approaches [1]. Despite this progress, there remains a notable gap in the availability of transparent models for time series analysis [8]. Transparency is particularly crucial in scenarios where model predictions have real-world implications, such as determining treatment strategies or triaging in healthcare settings [9]. Developing interpretable models is essential for ensuring accountability and promoting equity in high-stakes applications.

Challenges of time series forecasting

Time series forecasting is widely applied across various fields to analyze highly heterogeneous datasets that often include both static and dynamic (non-static) features with complex interdependencies [10]. Traditional models, such as recurrent neural networks (RNNs) and transformers, are capable of processing heterogeneous input features. However, these models lack transparency, as they primarily focus on predicting the values of a time series at specific points without providing insights into the underlying trends.

To address the need for greater interpretability, the TIMEVIEW model employs B-spline-based interpolation to visualize trends in forecasting and analyze the effects of input perturbations [8]. While this approach enhances transparency, the original TIMEVIEW model is limited to predictions based solely on static features. In real-world scenarios, however, datasets frequently contain both static and dynamic features. For instance, in intensive care unit (ICU) settings, both types of data are readily available: static features such as height, weight, age, and gender, alongside time-series data such as heart rate, respiratory rate, and oxygen saturation [11].

Problem Formulation

To illustrate the problem formulation, we consider the ICU dataset used in the study. Each sample consists of:

- **Static features**: $\mathbf{x} \in \mathbb{R}^M$, where $M \in \mathbb{N}$ represents the number of static features.
- **Time-series data**: n time-series $\mathbf{z}_i(t) \in \mathbb{R}^{N_d}$, where $i = 1, \ldots, n$, $N_d \in \mathbb{N}$ is the number of measurements at discrete time points $t \in \mathbb{R}^{N_d}$, and $n \in \mathbb{N}$ is the number of time-series.

For the ICU dataset $M = 2$ and $n = 2$:

- The static features include:
  - Gender (binary: 0 or 1),
  - Age (range: 0–90).
- The time-series data include:
  - Respiratory rate and oxygen saturation, with each series consisting of $N_d = 50$ data points at discrete time points.

The target series, $\mathbf{y} \in \mathbb{R}^{N_d}$, represents noisy samples from an underlying true continuous trajectory y_*. In this dataset, the target series corresponds to the heart rate measurements at the same 50 discrete time points as the respiratory rate and oxygen saturation series. Alternatively, the target could also have different timepoints, such as predicting the next 50 data points of the respiratory rate. Note that time-series of varying lengths can be resampled to ensure consistent input lengths.

Given a dataset $\mathcal{D} = \{\mathbf{x}^{(d)}, \mathbf{z}^{(d)}(t), \mathbf{y}^{(d)}\}_{d=1}^{D}$, where $D \in \mathbb{N}$ is the number of samples, the goal is to develop a model f that maps:

$$f : \{\mathbf{x}^{(d)}, \mathbf{z}^{(d)}(t)\} \mapsto \hat{y}(t),$$

whereas $\mathbf{z}^{(d)}(t)$ denotes the $n \times N_d$ matrix of time-series measurements for the d-th sample. The model is trained to minimize the expected value of the following loss function:

$$\mathcal{L} = \frac{1}{T} \int_0^T \left(\hat{y}(t) - y(t)\right)^2 dt,$$

which corresponds to minimizing the mean squared error (MSE) between the predicted trajectory $\hat{y}(t)$ and the underlying true trajectory $y_*(t)$ over the time interval $[0, T]$. This integral is approximated considering all discrete timepoints of the two series.

For the ICU dataset, the goal is to develop a model that predicts heart rate with the lowest mean squared error over a given time interval, based on the age and gender of the patient as well as the respiratory rate and oxygen saturation measurements over the same time window.

Contributions

Based on the work of Kacprzyk et al. [8], we introduce a modified version of the TIMEVIEW model, referred to as mTIMEVIEW. This enhanced model is designed to predict time series data from both static and dynamic (non-static) features. Importantly, mTIMEVIEW preserves the transparency and interpretability of the original TIMEVIEW framework, while extending its capabilities to better handle real-world scenarios with diverse, heterogeneous input data. This makes mTIMEVIEW more adaptable and applicable to a broader range of practical use cases.

2 Methods

mTIMEVIEW

To incorporate both static and dynamic (non-static) features, the TIMEVIEW model [8] was modified to accommodate the updated input structure. Since the model's output remains unchanged, only the encoder portion required adaptation. As depicted in Figure 1, the modified architecture replaces the original encoder's fully connected layer with a multi-layer perceptron (MLP). This MLP receives input from two distinct components: a fully connected layer that encodes the static features and a recurrent neural network (RNN) that independently processes each input time series. This design enables the encoder to combine static and temporal information, broadening the model's applicability while retaining its transparency and interpretability. The model's hyperparameters were fine-tuned across 10 tasks before training. For all tasks, the model was trained using a batch size of 64 for 100 epochs, and the number of basis functions was fixed at 9, consistent with the methodology of the original TIMEVIEW model.

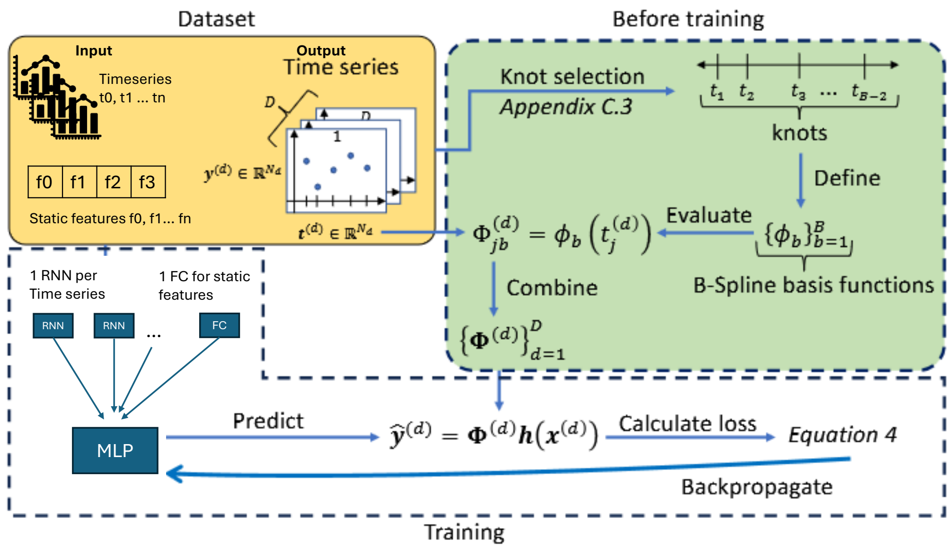

Figure 1: Improved model layout for timeseries forecasting with static and non-static inputs. Figure adapted from Kacprzyk et al. [8]

Baseline transformer

As a baseline, we implemented a compact transformer model, chosen for its versatility in processing both static and dynamic (non-static) features for time series prediction. The transformer architecture was configured with two attention heads, a 10-dimensional embedding size, and two layers. The embedding dimension was selected to align with the complexity level of the 9 basis functions used in the TIMEVIEW model, providing a fair basis for comparison. The model was trained for 100 epochs using a batch size of 64 and a learning rate of 0.01. To efficiently leverage both static and dynamic inputs, the static features and time series data were embedded independently. These separate embeddings were then concatenated and fed into the transformer's encoder.

Datasets

To evaluate the novel encoder architecture, new datasets were required because those provided by Kacprzyk et al. [8] were limited to predictions based solely on static features. We generated three synthetic datasets and utilized a publicly available dataset of vital sign measurements from intensive care unit (ICU) patients to comprehensively test our approach:

- **Sine Dataset**
  - This dataset consists of randomly generated sine functions with parameters shifted in amplitude and frequency $M = 2$ and $n = 1$.
  - The task requires the model to predict the new time series based on the original time series and the shift parameters.

- **Gaussian Dataset**
  - This dataset involves shifting the mean and variance of randomly generated normal distributions ($M = 2$ and $n = 1$).
  - The model must predict the resulting time series after these shifts.
- **Exponential Dataset (Exp)**
  - This dataset combines a random exponential series and a sine series as inputs.
  - A binary parameter specifies whether the two functions should be added or subtracted ($M = 1$ and $n = 2$).
  - The model's task is to predict the resulting time series based on these inputs.
- **Real-world ICU Dataset**
  - We used the publicly available BIDMC dataset [12] of ICU patient data containing time series of respiratory rate and oxygen saturation, along with static features such as age and gender ($M = 2$ and $n = 2$).
  - The model's task is to predict the resulting heart rate measurements based on these inputs.

For each synthetic dataset, we generated 500 random samples, with each time series having a fixed length of 25. The BIDMC dataset contained recordings from 50 patients. One recording was excluded due to missing age and gender information. The remaining 49 recordings were resampled to produce 10 samples per patient, each with a time series length of 50. Following the methodology of Kacprzyk et al. [8], we applied a [0.7, 0.15, 0.15] split for training, validation, and testing datasets. Model performance was evaluated using the mean squared error (MSE) between the predicted and ground truth time series for each dataset.

3 Results and Discussion

Both mTIMEVIEW and the transformer demonstrate strong performance on the synthetic datasets (see Table 1), with the transformer slightly outperforming mTIMEVIEW. However, the difference in performance on these synthetic datasets is minimal. When applied to the ICU dataset, both models achieve very low relative errors, which is notable given that the heart rate for most patients typically ranges between 80 and 90 bpm. Furthermore, both models comfortably exceed the minimum accuracy requirements set by the American National Standards Institute for heart rate monitoring, which mandate a mean absolute difference of less than 5 bpm or 10% of the signal [13].

While the performance of the two models is comparable, there are significant differences in computational efficiency. On a CPU, mTIMEVIEW requires nearly 50 times more computation time than the transformer (see Table 2). However, it is uncertain how these results would change when utilizing a GPU, which may considerably reduce mTIMEVIEW's computational overhead. Despite the increased runtime, the trade-off for enhanced model transparency may often justify the additional time and cost, particularly in applications where interpretability is critical.

Table 1: Average mean squared error (MSE) of the two models across the four datasets, along with the standard deviation calculated over five different random seeds.

Model	ICU dataset (bpm)	Sine dataset	Exp dataset	Gaussian dataset
mTIMEVIEW	3.63 ± 0.08	0.10 ± 0.00	0.04 ± 0.03	0.00 ± 0.00
Transformer	3.06 ± 0.05	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00

Previous work on time series forecasting has predominantly relied on black-box models such as transformers and recurrent neural networks (RNNs). In contrast, transparent models aim to provide interpretable representations of data, often involving methods that identify closed-form expressions, such as PySR [14] and SINDy [15]. More recently, Kacprzyk et al. introduced transparent models for time series prediction based on static features [8]. Their approach not only adheres to transparency standards but also offers an innovative framework for examining the underlying dynamics of predictions, going beyond merely predicting absolute values.

Table 2: Average training times of the two models across the four datasets, along with the standard deviation calculated over five different random seeds. Note that this training was done on a CPU and is that not necessarily representative for real-world deployment.

Model	ICU dataset (s)	Sine dataset (s)	Exp dataset (s)	Gaussian dataset (s)
mTIMEVIEW	2631	2107	2436	1899
Transformer	67	47	53	46

However, several limitations remain. As highlighted by the authors, restricting the model to static features significantly limits its applicability. Additionally, the model currently supports only a single type of basis function (cubic splines) which are well-suited for modelling smooth functions that vary gradually over time. This limitation poses challenges for accurately representing rapidly changing time series, such as voltage fluctuations during an action potential in a nerve cell. Such time series, characterized by sharp transitions and rapid dynamics, are unlikely to be adequately captured by cubic splines, reducing the model's utility in these scenarios. Expanding the range of supported basis functions and incorporating dynamic features could address these shortcomings and broaden the model's applicability.

To leverage the transparency strengths of the B-spline approach in mTIMEVIEW, the visualization platform would need to be enhanced to allow modifications to both static and dynamic (non-static) features. Manipulating time series data with a single slider can be challenging, but several user-friendly options could be introduced. For instance, users could adjust the mean, apply a linear trend, smooth the function with moving averages, or even draw a custom function, which could then be numerically approximated. These features would enhance the platform's flexibility and usability, making it more adaptable to complex data exploration and analysis.

Validation of the model should be extended to more complex and diverse datasets, which will require significant computational resources, particularly GPUs. A suitable example for such validation is the Early Prediction of Sepsis from Clinical Data dataset available on PhysioNet [16]. This dataset includes both static and dynamic features, such as heart rate and respiratory rate, which could be predicted using a combination of static metrics and dynamic metrics. Additionally, a risk function for sepsis onset could be calculated using methods like a Random Survival Forest, similar to the approach used in the flchain dataset by Kacprzyk et al. [8]. The mTIMEVIEW model would also have to be benchmarked against versions of the other models used in the original paper.

4 Conclusion

We present an extension to the TIMEVIEW system developed by Kacprzyk et al. [8], enabling transparent time series forecasting with both static and non-static features. The model demonstrates performance comparable to a small transformer across three synthetic datasets and one real-world dataset. However, more complex and diverse datasets are required to thoroughly evaluate the capabilities of the developed time series models. Addressing this limitation is beyond the scope of this study due to time and computational resource constraints.

Future work will focus on enhancing the TIMEVIEW visualization tool to better illustrate the impact of modifying time series inputs. Additionally, more advanced encoder architectures, such as transformers, will be explored to improve model performance. Comparative analyses with other time series forecasting methods, including symbolic regression and tree-based models where time is incorporated as a feature, will also be conducted to provide a broader assessment of the model's effectiveness.

5 Code and Data availability

The dataset analysed during the current study is available in the Physionet repository, `https://physionet.org/content/bidmc/1.0.0/bidmc_csv/#files-panel`. The underlying code for this study is available on GitHub and can be accessed via this link `https://github.com/Sr933/mTIMEVIEW`.

References

1. Casolaro, A., Capone, V., Iannuzzo, G. & Camastra, F. Deep Learning for Time Series Forecasting: Advances and Open Problems. *Information* **14,** 598. ISSN: 2078-2489. `http://dx.doi.org/10.3390/info14110598` (Nov. 2023).
2. Craddock, J. M. The Analysis of Meteorological Time Series for Use in Forecasting. *The Statistician* **15,** 167. ISSN: 0039-0526. `http://dx.doi.org/10.2307/2987390` (1965).
3. Li, D., Jiang, F., Chen, M. & Qian, T. Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks. *Energy* **238,** 121981. ISSN: 0360-5442. `http://dx.doi.org/10.1016/j.energy.2021.121981` (Jan. 2022).
4. Zeroual, A., Harrou, F., Dairi, A. & Sun, Y. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons & Fractals* **140,** 110121. ISSN: 0960-0779. `http://dx.doi.org/10.1016/j.chaos.2020.110121` (Nov. 2020).
5. Lim, B. *Forecasting Treatment Responses Over Time Using Recurrent Marginal Structural Networks* in *Advances in Neural Information Processing Systems* (eds Bengio, S. *et al.*) **31** (Curran Associates, Inc., 2018). `https://proceedings.neurips.cc/paper_files/paper/2018/file/56e6a93212e4482d99c84a639d254b67-Paper.pdf`.
6. Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing* **90,** 106181. ISSN: 1568-4946. `http://dx.doi.org/10.1016/j.asoc.2020.106181` (May 2020).
7. Lange, H., Brunton, S. L. & Kutz, J. N. From Fourier to Koopman: spectral methods for long-term time series prediction. *J. Mach. Learn. Res.* **22.** ISSN: 1532-4435 (Jan. 2021).
8. Kacprzyk, K., Liu, T. & van der Schaar, M. *Towards Transparent Time Series Forecasting* in *The Twelfth International Conference on Learning Representations* (2024). `https://openreview.net/forum?id=TYXtXLYHpR`.
9. Shick, A. A. *et al.* Transparency of artificial intelligence/machine learning-enabled medical devices. *npj Digital Medicine* **7.** ISSN: 2398-6352. `http://dx.doi.org/10.1038/s41746-023-00992-8` (Jan. 2024).
10. Wang, Y. *et al.* MTHetGNN: A heterogeneous graph embedding framework for multivariate time series forecasting. *Pattern Recognition Letters* **153,** 151–158. ISSN: 0167-8655. `http://dx.doi.org/10.1016/j.patrec.2021.12.008` (Jan. 2022).
11. Sauer, C. M. *et al.* Systematic Review and Comparison of Publicly Available ICU Data Sets—A Decision Guide for Clinicians and Data Scientists. *Critical Care Medicine* **50,** e581–e588. ISSN: 0090-3493. `http://dx.doi.org/10.1097/CCM.0000000000005517` (Mar. 2022).
12. Pimentel, M. *et al. BIDMC PPG and Respiration Dataset* 2018. `https://physionet.org/content/bidmc/`.
13. Institute, A. N. S. *Cardiac monitors, heart rate meters, and alarms*
14. Cranmer, M. *Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl* 2023. `https://arxiv.org/abs/2305.01582`.
15. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **113,** 3932–3937. ISSN: 1091-6490. `http://dx.doi.org/10.1073/pnas.1517384113` (Mar. 2016).
16. Reyna, M. *et al. Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019* 2022. `https://physionet.org/content/challenge-2019/`.